Automatic Sleep Classification with Machine Learning

Alexander Malafeev

University of Zurich[UZH]

Automatic Sleep Classification with Machine Learning

Dissertation

zur

Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Alexander Malafeev

aus

Russland

Promotionskommission

Prof. Dr. Peter Achermann (Vorsitz und Leitung der Dissertation)
Prof. Dr. Kevan Martin
Prof. Dr. Thomas König

Zürich, 2018

Summary

Sleep is ubiquitous in nature. Humans spend a third of their lives sleeping. And yet, despite all the recent advances in the field, we still don't know the purpose of sleep. However, sleep disorders are detrimental for health and quality of life and insomnia is one of the most common sleep disorders. Excessive daytime sleepiness or insufficient sleep decreases cognitive performance and may cause accidents. These facts suggest that understanding sleep and its regulation is very important.

In the first chapter I summarized the most recent hypotheses on the purpose of sleep; then I described the gold standard of assessing sleep – polysomnography (PSG), sleep stages and particularly the electroencephalogram (EEG). Furthermore, an overview of machine learning tools is provided as they will be used to classify sleep stages and detect microsleep episodes.

In the second chapter we implemented and tested 14 simple artifact detection methods and conducted a thorough analysis of their performance on two datasets, one comprised sleep recordings of healthy young subjects, the other one data recorded in patients with hypersomnia and narcolepsy. We found that clean average EEG power density spectra can be obtained using very simple methods. We got the best performance of artifact detection with thresholding slope, power in high frequency (25-90 Hz or 45-90 Hz) and the residual errors of an autoregressive model fitted to the EEG. It is not surprising that the power in high frequency range was a good predictor of an artifact as muscle artifacts are characterized by the power in high frequency range. Most methods showed good sensitivity. However, since we had chosen fixed false

positive rate (FPR) of 10%, we excluded on average 16.3% of the epochs whereas experts excluded on average only 7% of the epochs. Our approach seemed reasonable as it leaves enough data for subsequent analyses.

The main chapter (third chapter) describes the developed automatic sleep scoring algorithms. Scoring rules are complex and to some degree subjective. Despite the fact that human brain has superb image recognition abilities, sleep scoring is a difficult task. Thus, it is not possible to just program an algorithm which implements the scoring rules for sleep. Such a problem can be addressed however, with modern machine learning methods. Such algorithms learn from the examples which have already been analyzed by an expert. With these techniques we don't even need to know how to score sleep stages ourselves, we just need examples of experts. We developed several algorithms ranging from basic machine learning tools to deep artificial neural networks. First, we engineered 20 features derived from EEG, EOG and EMG data. This process reduces the dimensionality of our data and makes classification of the data easier. We employed a random forest (RF) classifier in conjunction with a Hidden Markov Model (HMM) or a moving median filter (MF) to smooth the data. Alternatively, we applied artificial neuronal networks (ANN), Long-Short Term Memory (LSTM) networks, designed to handle time series to classify the data. We used our engineered features as input for these networks. Finally, we employed deep convolutional neural networks (CNNs) in combination with LSTM networks. Such algorithms (CNN-LSTM networks) work with raw data and do not require engineered features. We used the F1 score, a performance measure of multi-class data which takes both specificity and sensitivity into account, to evaluate the quality of the automatic scoring. We achieved a sleep stage classification quality close to the human expert in recordings of healthy subjects, with F1 scores above 0.8 for all stages except

for stage 1. Stage 1 is difficult to score for a human scorer as well. F1 scores of stage 1 were slightly above 0.4 for most our methods, like the interscorer performance. Our methods trained on healthy participants performed slightly worse on the patient data than on the data of healthy subjects when they were trained only on the data of healthy subjects. However, the performance of the ANNs was better than RF in this case. Performance on the patient data improved when patient data were included into the training. We demonstrated that the methods which incorporate the temporal structure generally perform better. Further, the methods relying on the raw data performed slightly better than the feature-based methods. We think that we could not use the whole potential of ANNs due to the scarcity of the training data.

Using these algorithms, we may score sleep fully automatically and analyze big amounts of data very quickly. Our CNN-LSTM network produced good results using just a single EEG channel. This was an unexpected result as we assumed that reliable detection of REM sleep would require EOG and EMG data. Such networks would allow on-line scoring of data recorded with portable devices.

The fourth chapter is dedicated to the automatic detection of microsleep episodes (MSE). MSE are very short sleep fragments lasting 3 to 15 s. They often occur in sleep deprived people, in individuals who had insufficient sleep or under boring or monotonous conditions, and in patients with excessive daytime sleepiness. We engineered features and applied basic machine learning methods (support vector machine, random forest) to detect MSE. In a preliminary step we demonstrated that the methods work and reached very good specificity (0.99) and good sensitivity (0.74). Future improvement of MSE detection algorithms should include the temporal

structure of the data, for example using LSTM neural networks. In summary, our preliminary analysis provides proof of concept that automatic detection of MSE based on sleep EEG data is feasible.

All together, we could demonstrate that machine learning approaches perform well in detecting sleep stages and MSE.

The final chapter provides an outlook on further improvements and future steps to be taken.

Zusammenfassung

Schlaf ist etwas ganz Natürliches. Ein Mensch verbringt etwa ein Drittel seines Lebens schlafend. Doch trotz all der beeindruckenden Fortschritte in diesem Bereich ist die Funktion des Schlafs noch immer nicht bekannt. Wir wissen jedoch, dass Schlafstörungen sich nachteilig auf die Gesundheit und die Lebensqualität auswirken. Eine der häufigsten Schlafstörungen ist die Insomnie. Exzessive Tagesschläfrigkeit und ungenügender Schlaf verringern die kognitive Leistungsfähigkeit und können zu Unfällen führen. Diese Tatsachen zeigen, dass das Verständnis von Schlaf und seiner Regulation eine wichtige Rolle spielt.

Im ersten Kapitel dieser Arbeit habe ich die neuesten Hypothesen über die Funktionen des Schlafs zusammengefasst. Anschließend bin ich auf die Polysomnographie (PSG), den sogenannten Goldstandard in der Schlafdiagnostik, die Schlafstadien und insbesondere auf das Elektroencephalogramm (EEG) eingegangen. Ferner wird ein Überblick über die «Machine-Learning»-Methoden gegeben, da diese für die Klassifizierung der Schlafphasen und die Erkennung von Mikroschlaf-Episoden verwendet werden.

Im zweiten Kapitel haben wir 14 einfache Methoden zur Erkennung von Artefakten an zwei Datensätzen eingehend getestet. Der erste Datensatz umfasste Schlaf-EEG-Ableitungen von gesunden jungen Probanden, der zweite Daten von Patienten, die unter Hypersomnie und Narkolepsie leiden. Wir haben festgestellt, dass sich mittels dieser sehr einfachen Methoden qualitativ gute mittlere leistungsdichte Spektren des EEGs ergeben. Die besten Ergebnisse bei der Artefakterkennung haben wir durch die Begrenzung der Steilheit der EEG-Auslenkung, der hochfrequenten Leistung im EEG (25-90 Hz oder 45-90 Hz) oder Abweichungen (residuals) von autoregressiven Modellen,

mit denen die EEG-Signale modelliert wurden, erzielt. Es ist nicht verwunderlich, dass sich die hochfrequenten Komponenten als ein guter Prädikator für das Vorhandensein eines Artefakts erwiesen, da aus der Literatur bekannt ist, dass sich Muskelartefakte durch eben diese Komponenten auszeichnen.Die meisten Methoden wiesen eine gute Sensitivität auf. Allerdings wurden aufgrund der Tatsache, dass wir eine Falsch-Positiv-Rate von 10 % festgelegt haben, durchschnittlich 16,3 % der Epochen ausschlossen, während Experten lediglich durchschnittlich 7 % ausschlossen. Dies schien uns angemessen, da noch genügend Daten war für die nachfolgenden Analysen zur Verfügung standen.

Im Hauptkapitel (drittes Kapitel) werden die entwickelten Algorithmen zur automatischen Schlafstadienbestimmung dargelegt. Die Regeln zur Bestimmung der Schlafstadien sind komplex und zu einem gewissen Grad subjektiv. Trotz der Tatsache, dass das menschliche Gehirn über ausgezeichnete Fähigkeiten zur Bilderkennung verfügt, erweist sich die Stadienbestimmung selbst für einen Menschen als ein schwieriges Unterfangen. Demzufolge ist eine manuelle Programmierung eines Algorithmus, der Regeln zur Bestimmung der Schlafstadien umsetzt, nicht möglich. Dieses Problem kann mittels moderner «Machine-Learning»-Methoden angegangen werden. «Machine-Learning»-Algorithmen lernen aus Beispielen, die bereits von einem Experten klassifiziert wurden. Dank dieser Methoden sind keine Kenntnisse in Bezug auf die Auswertung der Schlafphasen erforderlich. Wir benötigen lediglich Beispiele von Experten. Wir haben mehrere Algorithmen entwickelt, von grundlegenden «Machine-Learning»-Methoden bis hin zu künstlichen neuronalen Netzen («deep learning»). Zunächst haben wir 20 sogenannte «Features» auf Grund von EEG-, EOG- und EMG-Daten bestimmt, wodurch die Dimensionalität der Daten

verringert und ihre Klassifizierung vereinfacht wurde. Anschließend haben wir den sogenannten «Random-Forest»-Algorithmus (RF) verwendet und ein «Hidden-Markov-Model» (HMM) sowie einen Medianfilter (MF) zur Glättung der Daten angewandt. Danach haben wir ein künstliches neuronales Netz zur Verarbeitung von Zeitreihen, das sogenannte «Long-Short-Term-Memory»-Netz (LSTM), verwendet. Als Input für diese Netze dienten unsere entwickelten Features. Zum Schluss haben wir «Convolutional Neural Networks» (CNNs) zusammen mit einem LSTM Netzwerk angewandt. Ein solcher Algorithmus (wir bezeichnen diesen Algorithmus als CNN-LSTM) arbeitet mit Rohdaten und bedarf keiner entwickelten Features. Wir haben das F1-Mass, eine Messgrösse die Spezifität sowie Sensitivität bei mehrfach Klassen einbezieht, verwendet um die Qualität der automatischen Stadienerfassung zu beurteilen. Die Ableitungen der gesunden Probanden wiesen mit allen genannten Methoden eine hohe Schlafphasenklassifikationsgüte auf, die der von Experten entsprach. Bei den gesunden Probanden erzielten wir für alle Schlafstadien, mit Ausnahme von Stadium 1, ein F1-Mass von über 0,8. Auch für einen Menschen erweist sich die Erfassung von Stadium 1 als ein schwieriges Unterfangen. Bei den meisten unserer Methoden belief sich das F1-Mass für Stadium 1 auf etwa 0,4 wie das auch für die Übereinstimmung zwischen Experten zutrifft. Wir haben gesehen, dass, wenn unsere Methoden ausschliesslich mit Daten gesunder Probanden trainiert wurden, die Qualität der Klassifizierung mit diesen Methoden bei Patientendaten etwas geringer war als bei den Daten gesunder Probanden. Jedoch erwiesen sich in diesem Fall die künstlichen neuronalen Netze als leistungsfähiger als der RF-Algorithmus. Mit der Einbeziehung der Patientendaten in das Training verbesserte sich auch die Qualität der Klassifizierung bei Patientendaten. Wir haben nachgewiesen, dass die Methoden, die zeitlichen Strukturen der Daten einbeziehen, im

Allgemeinen eine bessere Qualität aufwiesen. Darüber hinaus erwiesen sich die auf Rohdaten gestützten Methoden als leistungsfähiger als die Feature-basierten Methoden. Unsers Erachtens nach war es uns aufgrund der geringen Menge an Trainingsdaten nicht möglich, das Potenzial der künstlichen neuronalen Netze voll auszuschöpfen.

Diese Algorithmen ermöglichen uns eine voll automatische Schlafstadienerfassung sowie eine äusserst schnelle Analyse grosser Datenmengen vorzunehmen. Entgegen unserer Annahme, dass für eine zuverlässige Erfassung des REM-Schlafs sowohl EOG- als auch EMG-Kanäle erforderlich sind, erzielte unser CNN-LSTM-Netz unter Verwendung eines einzigen EEG-Kanals sehr gute Ergebnisse. Solche Netzwerke erlauben eine «on-line» Klassifizierung von Schlafdaten die mittels portablen Geräten erfasst werden.

Im vierten Kapitel wird die automatische Erfassung der Mikroschlaf-Episoden (MSE) behandelt. Bei MSE handelt es sich um kurze Schlaffragmente von 3 bis 15 Sekunden, die nach Schlafentzug oder ungenügendem Schlaf, während monotonen Tätigkeiten oder bei Patienten mit exzessiver Tagesschläfrigkeit auftreten können. Für die Erkennung von MSE haben wir Features entwickelt und grundlegende «Machine-Learning»-Methoden («support vactor machines», «random forest») angewandt. In einem ersten Schritt haben wir gezeigt, dass die Methoden funktionieren und eine sehr gute Spezifität (0,99) und eine gute Sensitivität (0,74) aufweisen. Die Algorithmen zur MSE-Erfassung können zukünftig durch Einbezug der zeitlichen Struktur der Daten, zum Beispiel durch die Verwendung eines LSTM Netzes, verbessert werden. Wir haben einen «proof of concept» geliefert, dass die automatische Erkennung von MSE mittels EEG möglich ist.

Insgesamt konnten wir nachweisen, dass sich «Machine-Learning»-Ansätze bei der Erkennung von Schlafphasen und MSE als äußerst leistungsfähig erweisen.

Im letzten Kapitel wird ein Ausblick auf weitere Verbesserungsmöglichkeiten und zukünftige Entwicklungsschritte gegeben.

Acronyms

A1	Electrode on the left mastoid (behind the ear)
A2	Electrode on the right mastoid (behind the ear)
AASM	American Association for Sleep Medicine
ANN	Artificial Neural Network
AR	Autoregression
AUC	Area under the curve
BSS	Blind Source Separation
C3	central EEG electrode on the left hemisphere
C3A2	EEG channel C3-A2
CNN	Convolutional Neural Network
DAE	Denoising Autoencoder
DNN	Deep Neural Network
ECG	Electrocardiogram
EEG	Electroencephalogram
EMG	Electromyogram
EOG	Electrooculogram
F3	frontal EEG electrode on the left hemisphere
FFT	Fast Fourier Transformation
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FPR	False Positive Rate
GD	Gradient Descent
GPU	Graphic Processing Unit
HMM	Hidden Markov Model
Hz	Hertz
ICA	Independent Component Analysis
KM	k-Means
L1	Manhattan norm
L2	Euclidian norm
LASSO	Least Absolute Shrinkage and Selection Operator
LOC	Left Ocular Channel
LSTM	Long-Short Term Memory
MC	Mean Crossing
MF	Median Filter
ML	Machine Learning
MSE	Mean Square Error
MSLT	Multiple Sleep Latency Test

MT	Movement Time
MWT	Maintenance of Wakefulness Test
N1	Sleep stage 1 (light sleep)
N2	Sleep stage 2
N3	Sleep stage 3 (deep sleep; SWS)
NLP	Natural Language Processing
NREM	Non Rapid Eye Movement
PCA	Principal Component Analysis
PSG	Polysomnography
REM	Rapid Eye Movement
RF	Random Forest
RNN	Recurrent Neural Network
ROC	Right Ocular Channel
ReLU	Rectified Linear Unit
SAE	Stochastic Autoencoder
SEF	Spectral Edge Frequency
SEM	Slow Eye Movement
SGD	Stochastic Gradient Descent
SOREM	Sleep Onset Rapid Eye Movement
SPC	Specificity
SSRI	Selective Serotonin Reuptake Inhibitors
SVM	Support Vector Machine
SWA	Slow Wave Activity
SWS	Slow Wave Sleep (stages 3 and 4)
SpO2	Blood oxygen saturation
TN	True Negative
TP	True Positive
TPR	True Positive Rate
VAE	Variational Autoencoder
ZC	Zero Crossing
fMRI	functional Magnetic Resonance Imaging
fs	sampling frequency
t-SNE	t-distributed Stochastic Neighbor Embedding
µV	microvolt

Acknowledgements

I relied on the help of many people during my PhD. First of all I acknowledge Prof. Dr. Peter Achermann. He gave me a chance to accomplish the PhD. Peter created comfortable and nourishing environment. Peter, I learned from you many things. Not only scientifically, but also on the personal level. Experience during this PhD tremendously changed many aspects of my thinking. I had a unique opportunity to develop my ideas in comfortable environment. I am very grateful to Prof. Dr. Alexander Borbély and Prof. Dr. Irene Tobler for teaching me a lot and helping out. This work would not be possible without help and support of my thesis committee members Prof. Dr. Peter Achermann, Prof. Dr. Kevan Martin and Prof. Dr. Thomas König.

I am extremely grateful to my friends Dr. Dmitry Laptev, Valentina Lapteva, Dr. Valery Vishnevsky, Alexander Kolesnikov, Lera Kolesnikova, Arseny Klimovsky, Nikolay Savinov, Alexey Gronsky and Elena Gronskaya and all other friends for being around and supporting me.

This work would not have been possible without contributions from my collaborators Dr. Dmitry Laptev, Dr. Stefan Bauer, Dr. Ximena Omlin, Dr. Aleksandra Wierzbicka, Dr. Adam Wichniak, Prof. Dr. Wojciech Jernajczyk, Prof. Dr. Robert Riener, Dr. Jelena Skorucak and Prof. Dr. Joachim Buhmann.

I am grateful to Jakub Michankow for the permission to use the photo he made as a cover picture (it was modified).

Dmitry Laptev, Valery Vishnevsky and Lilit Poghosyan helped me proofreading this thesis. Lilit helped me a lot with the English language in general. I am very thankful to Anna Neumann for teaching me German.

I am very thankful to all my office and group mates and all the members of the institute for help and support! I am grateful to Dr. Thomas Rusterholz, Ueli Wyss and Dr. Roland Dürr for technical support.

I am expressing the greatest gratitude to my parents for raising and supporting me!

Contents

1 Introduction

1.1 What is sleep

We all sleep, and we have a notion of what sleep is. Sleep may be defined on a behavioral level or based on electrophysiological (see further below).

A behavioral definition of sleep was developed by Piéron (Piéron, 1913) and extended by Flanigan et al. (Flanigan Jr et al., 1974). According to behavioral definition, sleep is the state when animal is (1) immobilized, (2) chooses specific place to sleep, for example a nest, (3) has a characteristic body posture, (4) an animal can be quickly woken up, (5) the animal's arousal threshold is higher than in wakefulness, (6) sleep is homeostatically regulated. The requirement of homeostatic regulation was introduced by Irene Tobler (Tobler, 1984).

It has been shown that most studied species show clear signs of sleep, including drosophila (Hendricks et al., 2000), zebrafish (Zhdanova et al., 2001) and even C. Elegans (Raizen et al., 2008). However, there are certain species whose sleep is more questionable, for example, the bullfrog (Hobson, 1967).

It is well known that sleep deprivation in human leads to cognitive impairments (Kjellberg, 1977, Alhola and Polo-Kantola, 2007, Kerkhof and Van Dongen, 2010, McCoy and Strecker, 2011). It also affects the mood (Banks and Dinges, 2007). The death of animals after prolonged total sleep deprivation was observed in several species: rats (Everson et al., 1989), drosophila (Shaw et al., 2002) and cockroaches (Stephenson et al., 2007).

Interestingly, sleep deprivation can have a positive effect in depressed patients: it alleviates depression (Giedke and Schwärzler, 2002). However, the effect disappears after recovery sleep.

It seems that sleep is universal and essential. Despite the fact that the function of sleep is unknown, there are many theories addressing this question (Rechtschaffen, 1998, Mignot, 2008, Cirelli and Tononi, 2008).

1.2 Sleep theories

1.2.1 Energy conservation

One of the first hypotheses on the function of sleep was an idea that sleep might have been evolved due to a reduced energy consumption during this state (Walker and Berger, 1980). However, there is already a state of torpor which serves as a means of energy conservation and animals experience a sleep rebound after they come out of torpor (Heller and Ruby, 2004). Moreover, energy consumption is reduced only in NREM sleep, but not in REM sleep (Zhang et al., 2007).

1.2.2 Cellular maintenance

Another widely known hypothesis is a recovery hypothesis. It suggests that sleep is needed to recover cellular structures (Mackiewicz et al., 2007). Some studies, though, showed that sleep does not affect protein synthesis (Clugston and Garlick, 1982).

Vyazovskiy and Harris (Vyazovskiy and Harris, 2013) proposed a hypothesis that neurons have limited capacity to perform information processing and should undergo cellular maintenance to repair the “wear and tear” damage. Unless it happens, sleep might intrude into wakefulness to prevent permanent damage to neurons at cost of reduced performance during wakefulness. The authors suggested that maintenance can only be performed when neuron is disconnected from the network activity.

1.2.3 The memory and synaptic homeostasis hypothesis

It has been suggested that sleep is crucial for the information processing.

Tononi and Cirelli came up with the synaptic homeostasis theory (Tononi and Cirelli, 2006). New synapses are formed during wakefulness due to learning of new things. At some point, the ability of the brain to form new synapses saturates. Therefore, the net synaptic strength needs to be adjusted and decreases during sleep, particularly NREM sleep. According to the synaptic homeostasis theory, this is the crucial function of sleep.

A number of other studies have shown that sleep facilitates learning and memory consolidation (Karni et al., 1994, Stickgold, 2006, Born et al., 2006, Yoo et al., 2007).

It has also been observed that replay of the activations which had happened during wakefulness may occur during sleep (Pavlides and Winson, 1989, Ji and Wilson, 2007, Stickgold et al., 2001, Diekelmann and Born, 2010). This phenomena is called hippocampal replay.

1.2.4 Cleaning of "brain waste"

Recent studies conducted by Dr. Maiken Nedergaard and her colleagues showed that cerebral fluid flow dramatically increases during sleep (Xie et al., 2013, Iliff et al., 2012). The space between neurons enlarges and more fluid flows through these clefts. The proposed theory says that it helps to flush out toxins, particularly beta-amyloid. Brain does not have a lymphatic system and such a mechanism can be a substitution of the lymphatic system. So far increase in the cerebral fluid flow has been shown in animals, but not yet in humans.

1.3 Sleep disorders

Sleep is of a big interest for medicine since the prevalence of sleep disorders is tremendous. According to some studies (Bixler et al., 1979, Hersberger et al., 2006, Ohayon, 2002), up to 50% of population suffer from some kind of sleep disorders, mainly, insomnia.

It is clear that sleep affects health: people with disturbed sleep have increased risk of cancer and it damages immune system (Irwin, 2015). Moreover, reduced sleep duration leads to metabolic diseases such as type two diabetes (Copinschi et al., 2014). These patterns have been extensively studied in shift workers: they have higher risks of cancer, diabetes, depression and cardiovascular problems (Faraut et al., 2013, Marquié et al., 2014, Ramin et al., 2015). That is the reason why studies on how changes in sleep influence human health are of a great importance. The importance of such studies is still growing because sleep duration has been decreasing over the last several decades (Tinguely et al., 2014). As a further matter, sleepiness, sleep loss and excessive daytime sleepiness have been one of the causes of major industrial accidents (for example Chernobyl) and transportation (Rajaratnam and Arendt, 2001).

Another common sleep disturbance is obstructive sleep apnea (Force and Medicine, 2009). It affects people's well-being, causes excessive daytime sleepiness, which leads to accidents on transportation, and increases risks of developing chronic illnesses.

A lot of other sleep-related conditions require evaluation in a sleep laboratory. Hypersomnia and narcolepsy are among such disorders (Roth and Broughton, 1980). Both conditions are manifested in the excessive daytime sleepiness. Narcolepsy, for instance, can be manifested either in sleepiness only or in sleepiness combined with sleep attacks and cataplexy. During a

cataplectic attack, a person loses muscle tone and collapses; it mostly happens after experiencing strong emotions

All these conditions require evaluation for diagnosis, treatment and in some countries fitness to drive must be evaluated and is required for affected people in order to possess a driving license, particularly for professional drivers.

1.4 Sleep evaluation

Sleep evaluation is needed both in research and medicine. Research questions such as sleep regulation, sleep and health etc. require objective measures. The gold standard of sleep studies is polysomnography, which is a recording of several biosignals (including at least the first three signals) listed below.

1.4.1 Electroencephalogram (EEG)

The EEG is the most important state indicator for us. Electrodes on the scalp measure electrical field potential changes, which result from postsynaptic potential changes of pyramidal neurons in the cortex (Buzsáki et al., 2012).

The EEG measures the difference in the potential between two electrodes. One is placed in the area of the interest – on the scalp, the other one is the reference electrode. Common references in sleep research and medicine are the contralateral mastoids (behind the ear). The left mastoid electrode is named A1, the right one A2.

Other EEG electrodes are usually named with a letter and a number. Letter stands for the location: O-occipital, P-parietal, C-central, F-frontal, T-temporal. Number also reflects location according to the electrode placement

system “10-20 system” (Jasper, 1958) (Figure 1.1). Odd numbers stand for the left hemisphere, even ones for the right hemisphere and the index z indicates the midline.

In this way derivations are named after the two electrodes concerned, for example C3-A2 is the channel which is commonly used for scoring (Rechtschaffen and Kales, 1968). It means that the potential difference of the electric field is measured between C3 and A2.

Derivations can be referenced in other ways, for example F3-C3 or an electrode can be referenced to the average reference (mean of all electrodes in case of high-density EEG recordings), however we did not work with such references.

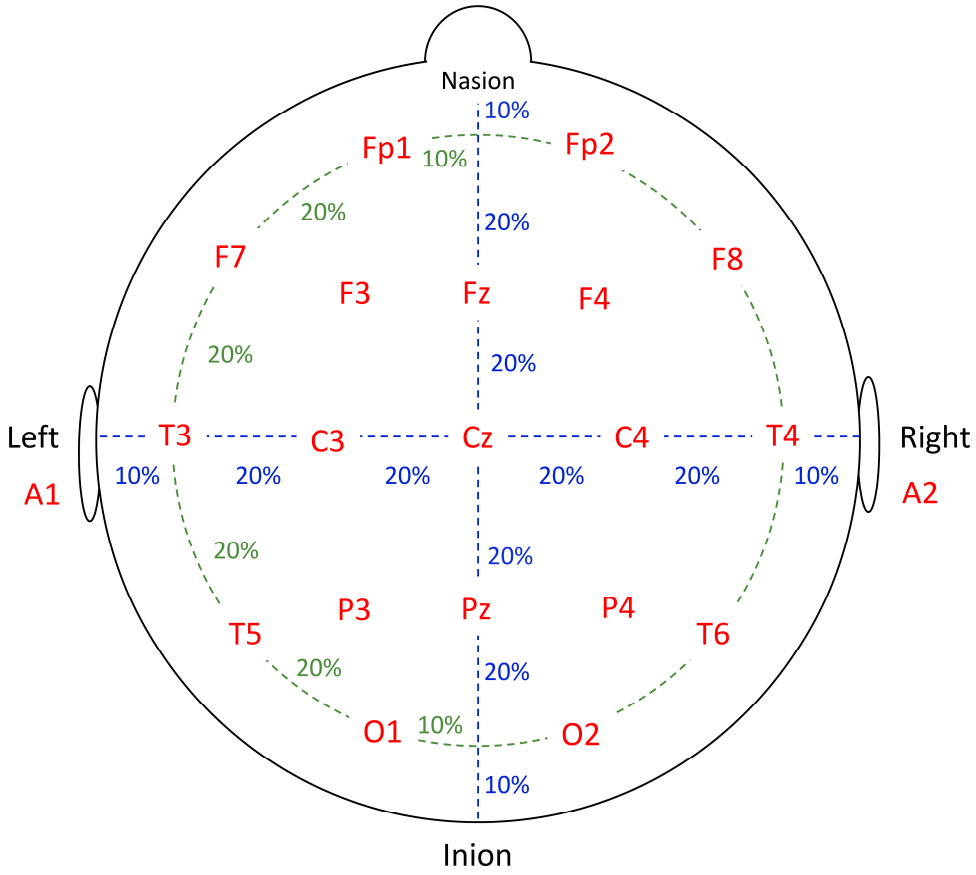


Figure 1.1. Electrode placement according to the 10-20 system. Modified from ”Bits of Sleep” (Borbély et al., 1998)

Neuronal activity creates oscillations of different frequencies in the EEG signal. One of the most widely known oscillations is alpha oscillation. It was discovered by Hans Berger (Berger, 1929). It is an oscillation with a frequency around 10 Hz. Alpha oscillations appear in relaxed wakefulness with closed eyes. When the subjects open their eyes, alpha oscillations generally disappear (alpha blocking).

Delta (slow) waves are oscillations in the frequency range of 0.5 – 4 Hz. They were discovered by Walter Grey (Walter, 1936). Slow waves are the marker of deep sleep (Rechtschaffen and Kales, 1968).

EEG power in the range 0.5-4.5 Hz is called slow-wave activity (SWA). SWA is homeostatically regulated and one can observe a rebound after sleep deprivation (Borbély et al., 1981, Borbély and Achermann, 1999).

Another important oscillation is a sleep spindle. This is a waxing and waning oscillation in the frequency range 12-14 Hz with a duration of 0.5 to 2 s (Rechtschaffen and Kales, 1968). Sleep spindles are a main property of the sleep stage 2 (see below).

1.4.2 Electrooculogram (EOG)

Electrodes located on the skin near the eyes record changes in the potential of electric field due to the eye movement. This change in the potential is caused by the fact that eye is a dipole (Marg, 1951). Eye movements are essential to score sleep because every sleep stage has distinct patterns of eye movements.

Usually two EOG channels are used. One electrode is located above the outer canthus (corner) of the left eye. This channel is called Left Ocular Channel (LOC). It is usually referenced to one of the mastoids (A1 or A2). The

other electrode is located below the outer canthus of the right eye. This channel is called Right Ocular Channel (ROC).

Rapid Eye Movements (REMs) are one of the most prominent properties of REM sleep and they are manifested in anticorrelated deflections in LOC and ROC.

Eye blinks occur only during wakefulness and are helpful to distinguish this stage. Eye blinks also cause anticorrelated deflections in the LOC and ROC, but the shape of the signal is different. We observed that there are two different types of eye blinks: (1) when deflections in the LOC and ROC have the same amplitude and (2) when deflection in the LOC has a larger amplitude than the one in the ROC. We did not find reports on this phenomena in the literature. I think it happens because electrodes during eye blink record rather the activity of the muscles than the polarization of the eyeball. And there might be two distinct patterns of muscle activation for the eye blinks (see chapter 3.9.1).

1.4.3 Electromyogram (EMG)

Electrode placed on the muscle measures its activity, i.e. muscle tone. In sleep research it is common to record muscle tone from the electrodes located under the chin (submental EMG). Muscle tone is lower during sleep than during wakefulness. REM sleep is characterized by extremely low muscle tone, also known as REM sleep atonia (Jouvet et al., 1959, Rechtschaffen and Kales, 1968).

1.4.4 Breathing effort

Breathing effort is routinely measured by two belts, one located around the chest, the other is placed lower, measuring movement of the abdomen.

These belts detect changes in their length. Breathing in leads to a lengthening, breathing out to a shortening. In the case of obstructive sleep apnea, doctors see an increased breathing effort along with a drop in blood oxygen saturation and a cessation of airflow.

1.4.5 Snoring

Snoring can be recorded using a microphone. This signal is important in clinical setting. We did not use it in our studies.

1.4.6 Airflow

The airflow through the nose and mouth may be recorded using temperature sensor located below the nose. The exhaled air is warmer than the inhaled one.

1.4.7 Blood oxygen saturation (SpO2)

An important measure for screening patients for sleep apnea is blood oxygen saturation. It is usually measured at the fingertip. SpO2 drops when episodes of apnea occur.

1.4.8 Electrocardiogram

Electrocardiogram registers electric activity of the heart. It can be useful for sleep analysis and particularly for detection of sleep apnea events (Sivaranjni and Rammohan, 2016) and this signal may be used to correct cardiac artifacts in EEG channels.

1.5 Sleep stage scoring

A recording is minimally composed of the EEG, EOG and EMG signals. Further, these signals are evaluated by a professional. The recording is being split into 20- or 30-s long intervals, the scoring epochs. They are visually scored as wakefulness, sleep stages 1, 2, 3 and 4, and so-called paradoxical or rapid-eye movement (REM) sleep.

REM sleep was first found in cats by Rudolf Klaue in 1937 (Klaue, 1937), distinct electrical activity during dreaming was also observed by Loomis (Loomis et al., 1935, Loomis et al., 1937, Loomis et al., 1938). The first paper with the study of this state was published by Aserinsky and Kleitman (Aserinsky and Kleitman, 1953). They coined the term Rapid Eye Movement (REM) sleep.

At about the same time French scientist Michel Jouvet and his colleagues observed muscle atonia in cats accompanied by sporadic twitches. They called it “paradoxical sleep”. The paper (Jouvet et al., 1959) was published only some years after their discovery.

Sleep is being scored according to the scoring manuals. The first manual was published in 1968 by Rechtschaffen and Kales (Rechtschaffen and Kales, 1968). According to this manual, sleep was classified into wake, non-rapid eye movement (NREM) sleep stages 1, 2, 3, and 4, REM sleep and movement time (MT), i.e. when a subject moved and a signal was contaminated with movement artifacts. Stages 3 and 4 are considered as slow wave sleep (SWS, deep sleep).

In the novel scoring rules published by American Association of Sleep Medicine (Iber et al., 2007), basically SWS was named N3, no longer subdivided leading to the NREM sleep stages N1 to N3, and MT was abolished.

In my opinion, stage MT is important because otherwise it is not clear how to score such epochs contaminated by an artifact, especially when it

comes to automatic scoring algorithms. I noticed that these epochs were often recognized by an algorithm as wakefulness, which definitely makes sense. In this case, though, we have clear discrepancy between an expert and the computer. Experts usually score such epochs as the same stage as the surrounding sleep.

Examples of distinct EEG, EOG and EMG patterns in the different sleep stages are illustrated in Figure 1.2.

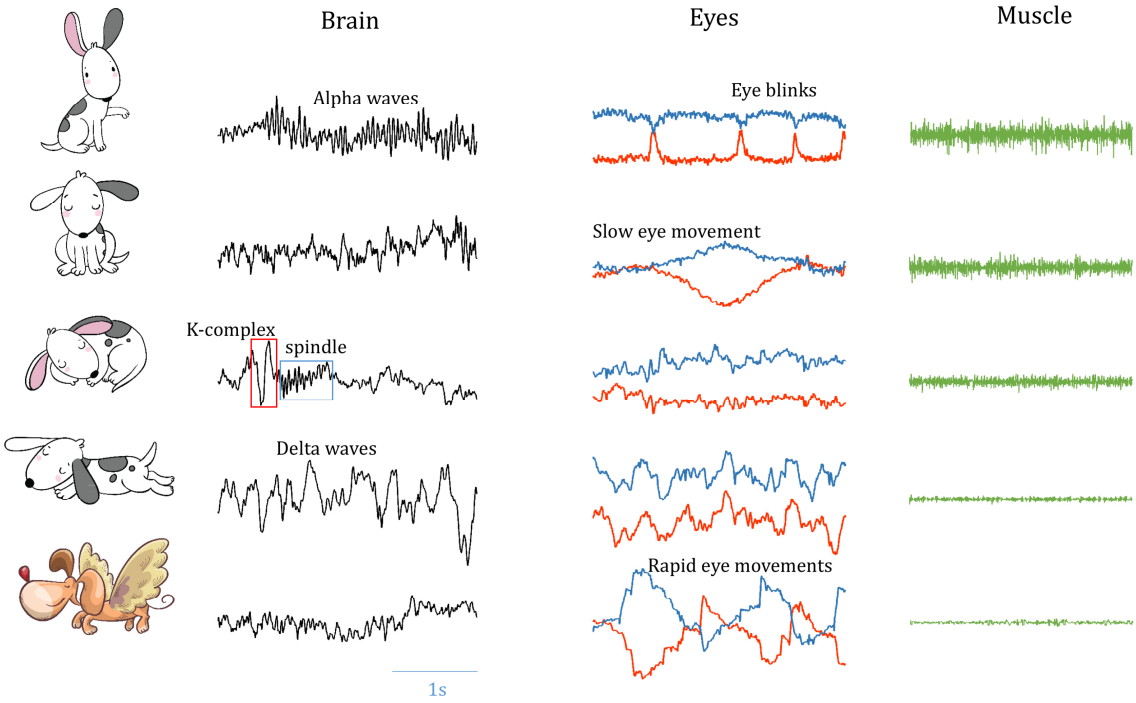


Figure 1.2. This figure shows examples of the EEG, EOG and EMG signals in the different sleep stages (from top to bottom: Wake, Stage 1, 2, 3, REM). Images from following sources were used: Natasha_Chetkova/Shutterstock; Alina Odryna/Shutterstock

In order to score a sleep recording, an expert splits the recording into consecutive into 20- or 30-s long intervals and assigns the stage based on the patterns the expert sees in the signals. This process is very time consuming, and, according to several studies (Danker-Hopfe et al., 2004, Penzel et al., 2013, Rosenberg and Van Hout, 2013, Younes et al., 2018), human experts are prone to make mistakes and have a lot of disagreement with each other. For

this reason, a number of attempts were undertaken to score sleep automatically. However, no standard has yet been established.

1.6 Quantitative EEG analysis: spectral analysis

Hypnograms and visual representation of PSG signals provide good overview of the sleep structure and can be used by medical doctors to diagnose many sleep disorders. For many applications it is not enough to have qualitative description of the data. Certain research and clinical questions can be better addressed using quantitative analyses. One of the most widely used quantitative measures of sleep are EEG power density spectra. Several important parameters can be derived from the spectra. Some of them are listed below and in Figure 1.3.

1.6.1 Slow wave activity (SWA)

First of all one can compute power in the low frequency range (0.75 - 4.5 Hz), called slow wave activity (SWA) which is a reliable marker of sleep homeostasis (Borbély and Achermann, 1999).

1.6.2 Sleep “fingerprint”

Average power spectra of distinct sleep stages are quite an interesting characteristic of sleep. It was shown that average spectra were very stable and may be considered as a sleep “fingerprint” (Lennox et al., 1945, Stassen, 1980, Buckelmüller et al., 2006, Bersagliere et al., 2018). Sleep EEG power density spectra were very similar in monozygotic twins (De Gennaro et al., 2008).

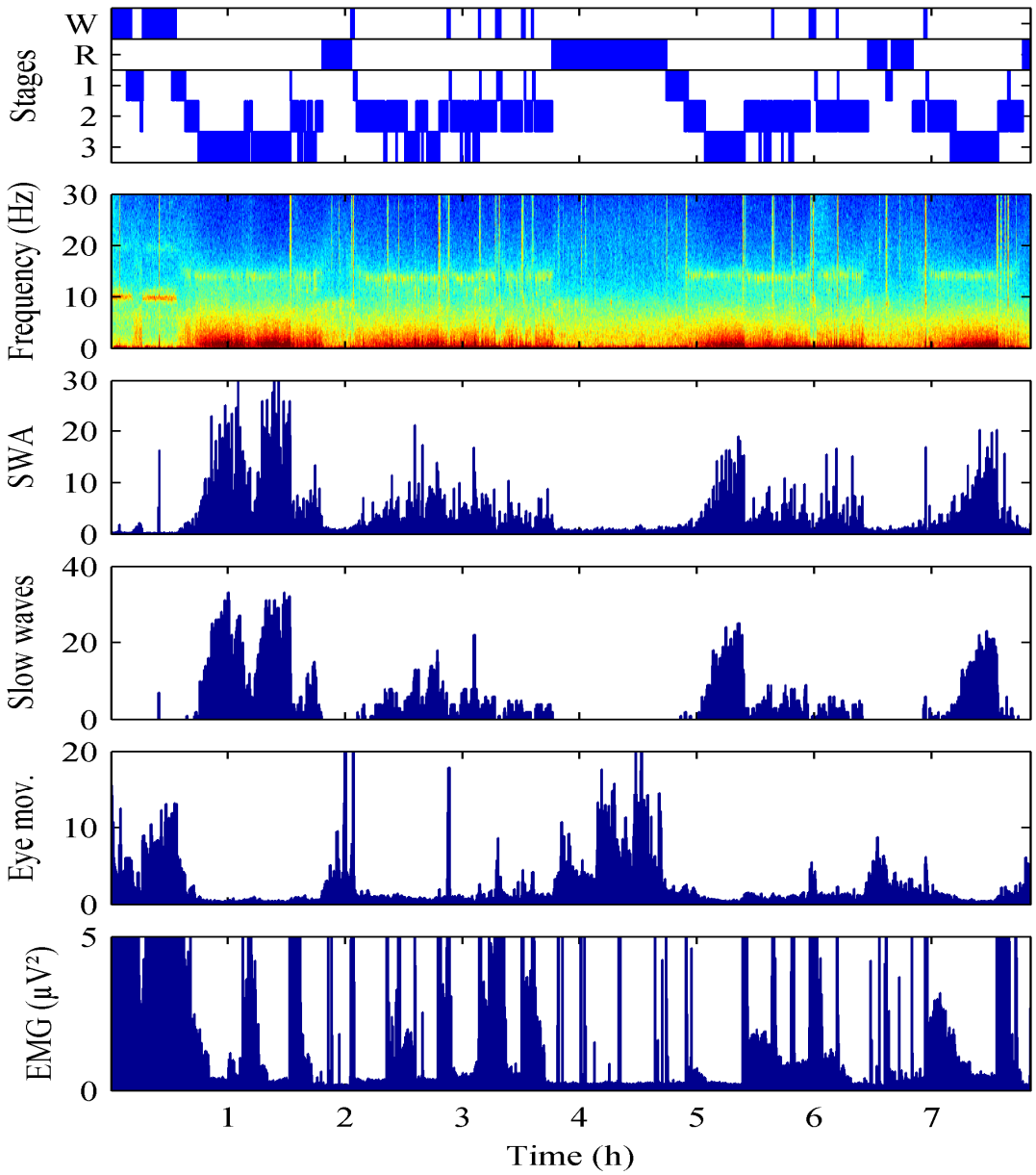


Figure 1.3. This figure shows the recording of a night of sleep and extracted quantitative parameters (features). Panel 1: sleep hypnogram; Panel 2: spectrogram; Panel 3: slow wave activity (SWA); Panel 4: the number of high amplitude slow waves per epoch; Panel 5: SWA of the ocular channel (LOC-ROC) divided by SWA of the EEG; Panel 6: Power in the chin EMG. The figure is from a conference abstract (Achermann et al., 2015)

1.6.3 Benzodiazepine

Certain drugs affect average power spectra. Benzodiazepines reduce slow wave activity and enhance spindle activity (Trachsel et al., 1990, Tobler et al., 2001). This is also true for Z-drugs (analog substances) (Brunner et al., 1991). Such changes in the power density spectra are very similar for the different drugs, also called the spectral signature of benzodiazepines and analogs. The example of the change of the power spectra, caused by three drugs, relative to placebo is illustrated in Figure 1.4 Trachsel et al., 1990, Brunner et al., 1991, Borbély et al., 1998).

1.7 Artifacts

Artifacts are detrimental for both quantitative spectral analysis and for automatic scoring. It is necessary to identify epochs with artifacts and exclude them from spectral analysis. It is very useful to perform it automatically.

In this work, we addressed both automatic artifact detection and automatic sleep scoring. A number of attempts to solve the problems of artifact detection have been made for some time now (Ktonas et al., 1979, Barlow, 1983, Barlow, 1984, Barlow, 1986, Bodenstein and Praetorius, 1977, Gotman et al., 1981, Durka et al., 2003) (D'Rozario et al., 2015, Coppieters't Wallant et al., 2016).

Fortunately, nowadays we can tackle these issues with novel methods as machine learning methods have advanced with an enormous pace. Artificial neuronal networks were proven to be superior to classical machine learning methods (decision trees (Safavian and Landgrebe, 1991), Support Vector Machines (SVMs) (Cortes and Vapnik, 1995), logistic regression etc.) for most types of data.

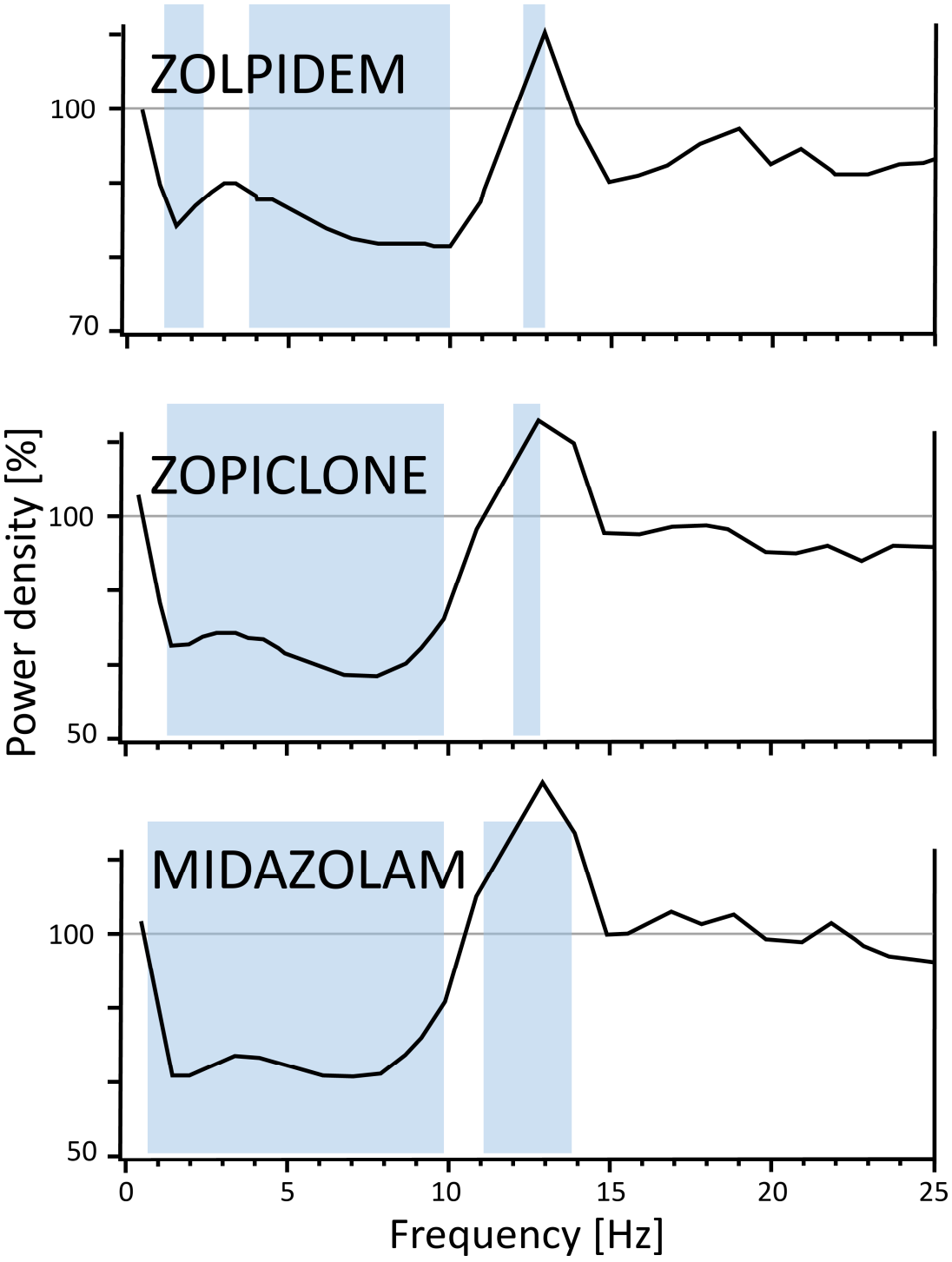


Figure 1.4. Effect of three sleep medications on the NREM sleep EEG power spectra. The change is relative to the placebo condition. Blue color covers frequency range with a statistically significant difference. Figure modified from (Borbély et al., 1998)

1.8 Machine Learning

A newly appeared branch of computer science, called Machine Learning, allows computers to learn how to label data without either directly programming the classification rules or even knowing them.

Machine learning can also be used to solve regression and clustering problems. If we want to assign labels to the data and we have the so-called training set, i.e. dataset with labeled examples available, it is a classification problem.

In case we do not have training set with labels we can perform a clustering. For example, we have data points in some space and we want to group them. The algorithm groups the data in a way that, for example, the sum of some metric (for example Euclidian distance) between the point and the center of a corresponding group is minimal. One of the most widespread algorithms to solve this problem is K-Means (Steinhaus, 1956). This algorithm arranges data into K clusters.

If we have examples of labeled data, we should use classification algorithms. The algorithm will learn statistical properties of the dataset and "understand" how to label new data points. Such type of learning from the data labeled by an expert is known as supervised machine learning.

1.8.1 K-means clustering

This is the most widely known clustering algorithm. We can describe every data point by a vector of the length d (dimension). For instance, we have patient data and we measured temperature and height, in this case d=2. We can plot our points in a two-dimensional plane (Fig. 1.5).

The idea is to split the feature space into k segments in a way that every segment contains a similar number of data points. The algorithm is iterative.

Let us choose k centroids randomly. They will be centers of our clouds of points. Then we will split the space into two parts with a line in a way that the distance from the line to both centroids is the same. In the case of multidimensional space it will be a multidimensional surface. We will assign the labels to the points that all the points on one of the sides belong to the class of corresponding centroid on the same side. Then we recalculate coordinates of centroids and repeat the procedure until it converges.

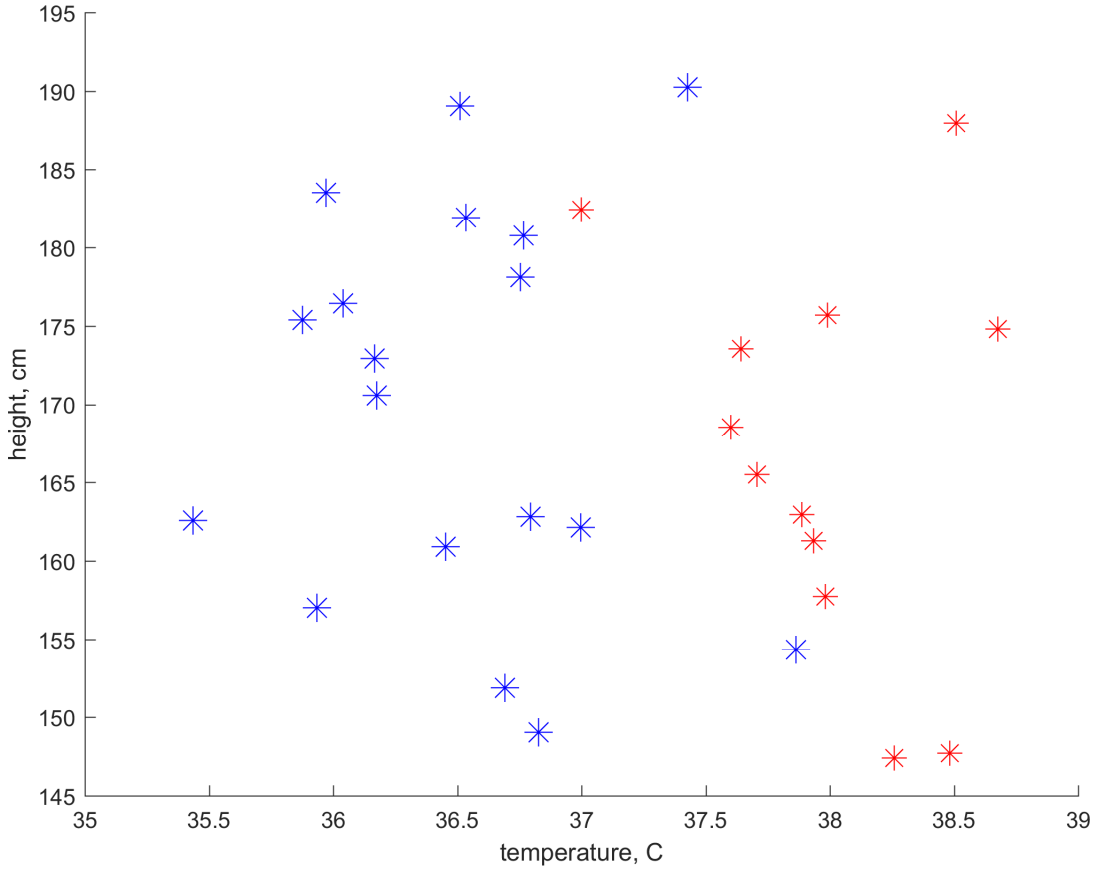


Figure 1.5. This figure shows an example of two-dimensional data of healthy and ill people

1.8.2 Logistic regression

The simplest approach to do a classification is the logistic regression (Cox, 1958). It is similar to simple linear regression but the value of the function belongs to the interval [0, 1]. The logistic function is shown below:

$$P(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \quad (1.1)$$

A subtype of a logistic function is a sigmoid function which is widely used in neural networks as an activation function.

1.8.3 Cost function

After we fit a linear regression, we usually use mean square error (MSE or L2 norm) to understand if the fit is good. Moreover, the fitting procedure is minimizing the MSE. Such a measure is called a cost function. It tells us how much mistakes cost. It does not necessarily have to be an MSE, it can be, for example, a sum of absolute values of errors (L1 norm). Cross-entropy is also a commonly used loss function for classification purposes (De Boer et al., 2005). Cross entropy provides a good measure of errors when our targets are discrete and we predict probabilities. Assume we have an epoch of sleep labeled as REM sleep. Then target probability is 1. And we predict probability p. If p is close to 1 then cross entropy loss is close to zero. If p is close to 0 then cross entropy loss is very big and it increases non-linearly because it is based on the logarithmic function.

1.8.4 The problem of overfitting

If we want to fit a straight line into a set of points, we use only two parameters and the result looks like the one on the Fig. 1.6 (top). The data used for fitting are represented by the blue dots. The red dots show new data points from the same distribution.The fitted line catches the trend but there is a certain discrepancy between the data points and the corresponding values of the linear function. We can add quadratic term, cubic term etc. to our function. In the end we can have a function which goes exactly through every

point (Fig. 1.6 middle). You can see that one red dot is very far from the fitted line in the middle panel. Despite that, the line goes exactly through every blue point where the error is 0. But if we add new points on the plot we can see that the errors for the new data may become large. This case is called overfitting. Our model was fitted to irrelevant noise in the dataset. This also means that our model has a high variance. On the contrary, the first model is too simple, or, in other words, it has bias. The trade-off between the too simple and too complex models is called bias-variance trade-off. A case of a good bias-variance trade-off is shown in the Fig. 1.6 (bottom). There we fitted a quadratic function.

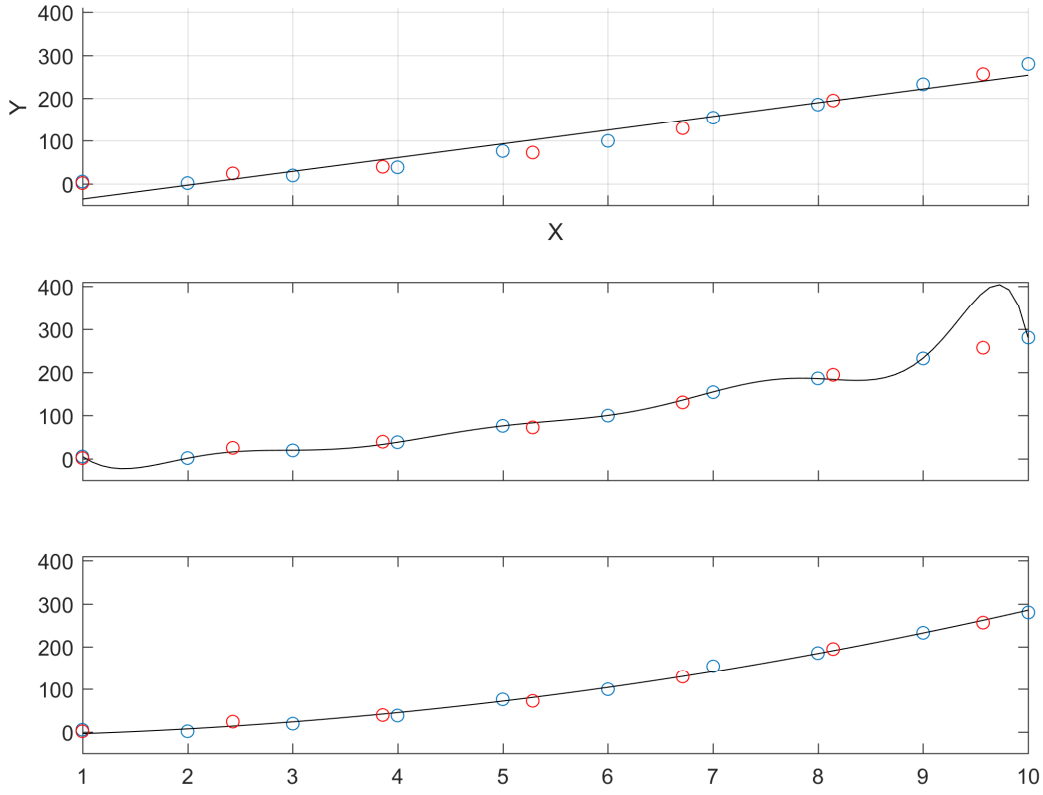


Figure 1.6. Fitting of polynomials of order 1, 20 and 2 to the data (blue circles). Red circles are new data points drawn from the same distribution

1.8.5 Regularization

The problem of overfitting can be addressed by regularization methods. The simplest method is to assign a penalty to the coefficients. It can be, for example, the sum of the squares of the polynomial coefficients multiplied by a regularization parameter λ.

The idea is that polynomial functions with larger coefficients can have larger variance in order to accommodate every data point. This type of regularization is called L2 or ridge regularization.

Another way is to use the sum of the absolute values instead of the squared ones. This is called L1 regularization or LASSO (Tibshirani, 1996).
L1 regularization assigns zeros to small coefficients. That's the most important property. It can be used for both feature selection and efficient removal of irrelevant features from the model.

1.8.6 Random Forest

Decision trees are widely used to solve classification problems (Morgan and Sonquist, 1963, Hunt et al., 1966, Breiman et al., 1984). A decision tree is a way to represent a set of rules. On every node of a decision tree a split on certain feature is being performed. The threshold is stored in the node. In order to assign a label to a data point, one has to go down the tree and compare the value of a corresponding feature to a threshold. Outcome of the comparison determines into which branch of the tree we go next. When the tree is traversed, we end up in the leaf which defines the corresponding label of a feature.

A decision tree is a good and simple method, but it is not robust to outliers. It means that outliers can affect the structure and performance of the tree. A way to overcome this problem is to use a set of trees: build number of

trees, an ensemble (random forest, RF). Each tree is built using a random subset of the data and a random subset of features (Ho, 1995, Breiman, 2001). Choosing a random subset of features is called feature bagging. While a tree is being grown, a feature for every new node is chosen in a way to maximize information gain.

This allows us to compute importance of features. In order to label new data point each tree assigns its own label. The eventual label is produced by "voting" of the trees. Probability of the point belonging to each class can also be computed. This probability is equal to the number of trees which assigned the data point to the corresponding class divided by the total amount of trees.

The RFs are superior to machine learning methods which use metric to compute distance between features because RFs are insensitive to renormalization, scaling and nonlinear monotonous transformations of features. In case of Support Vector Machine (SVM) (Cortes and Vapnik, 1995), features should be normalized. For example, we have temperature of a person in Celsius and height in millimeters. We want to classify ill and healthy people. Obviously temperature is an important feature and the height is irrelevant. Moreover, height is a kind of noise in this case. However, the variation of the height in millimeters will be much larger than the variation of the temperature in °C. Thus, distance between the two points will be driven by height, i.e. noise. For the RF, height will be irrelevant, it will quickly find out that temperature provides a larger information gain. Irrelevant features and noisy features can often be found in biological data.

The RF approach is very good for selecting relevant features due to feature bagging. One has to be aware that in case many correlated features are present, feature selection will not have a unique solution. Presence of correlated features is often the case in biology. We observed it in our data too.

It does not affect the quality of classification. Still, it would be a big issue if one wants to find out relevant features or establish causal relationships. As for correlated features, they can be decorrelated using principal component analysis (PCA) (Pearson, 1901).

Among other advantages, the RF classification has is the ability of learning complex rules. It also requires less hyperparameters (only the number of trees) and performs well in a wide range of this parameter. Oshiro et al. (Oshiro et al., 2012) studied the performance of RF on several datasets and found that the performance saturated at number of trees 64 or 128. Of course for different applications it may vary.

1.8.7 Boosting (Chen and Guestrin, 2016)

Imagine the following situation. You drop matches out of the box on the table. Then you ask a friend to estimate how many matches are on the table. The answer is unlikely to be precise. On the other hand, if you ask many friends independently and average their answers, you will get a good estimate of amount of matches. The same idea can be applied to classification: if we have a bunch of weak classifiers, we can average the outcome of classification, and the result will be better than the result of each of these classifiers. This method is called boosting and RF classification is an example of such a method.

1.8.8 Artificial neural networks

Modeling a neuron *in silico has* always been a fascinating thing to do. A lot of models had been proposed (Farley and Clark, 1954, Rochester et al., 1956) which were early on used for data classification (Rosenblatt, 1958). These models were eventually extended. And this, in its turn, has led to the

introduction of multilayer neural networks (Widrow and Hoff, 1960). These networks are now called artificial neural networks (ANNs).

ANNs are comprised of interconnected neurons (an example of an ANN is illustrated on the Figure 1.7). Each neuron calculates the weighted summation of the inputs. Weights are the parameters of a neuron. Weights shall be adjusted during training of the network. It has become possible to train large networks since the backpropagation (Werbos, 1974) algorithm was invented. The backpropagation algorithm helps to compute gradients. Weights are modified using the gradients by gradient descent algorithm or, for example, the Adam (Adaptive moment estimation) algorithm (Kingma and Ba, 2014).

In order to solve the problem of image recognition, artificial neural models were proposed. The very first work of Fukushima et al. (Fukushima and Miyake, 1982) was inspired by studies of Hubel and Wiesel (Hubel and Wiesel, 1959). Fukushima's algorithm is the first algorithm which resembled a Convolutional Neural Network (CNN).

The CNN in its modern form was discovered and popularized later by LeCunn et al. (LeCun et al., 1989) and Waibel et al. (Waibel et al., 1989). As the name suggests, such ANNs perform a convolution of input data (image) with a set of filters. These filters are adjusted during training. Convolution can be described in the following way: a window with some picture is moved across the input image. The picture on the window is being compared with the underlying part of the input image and the degree of similarity between the window and the underlying picture for every position of the window on the image is determined. This degree of similarity is calculated as a plain matrix multiplication between the window and the underlying part of the input. These adjusted filters (windows) result in a kind of feature extraction. It is also

important to mention that CNNs have already been successfully applied to one-dimensional signals, namely, to EEG recordings (Cecotti and Graeser, 2008, Mirowski et al., 2008). Cecotti et al. (Cecotti and Graeser, 2008) detected evoked potentials in the EEG with deep learning and Mirowski et al. (Mirowski et al., 2008) worked on seizure prediction.

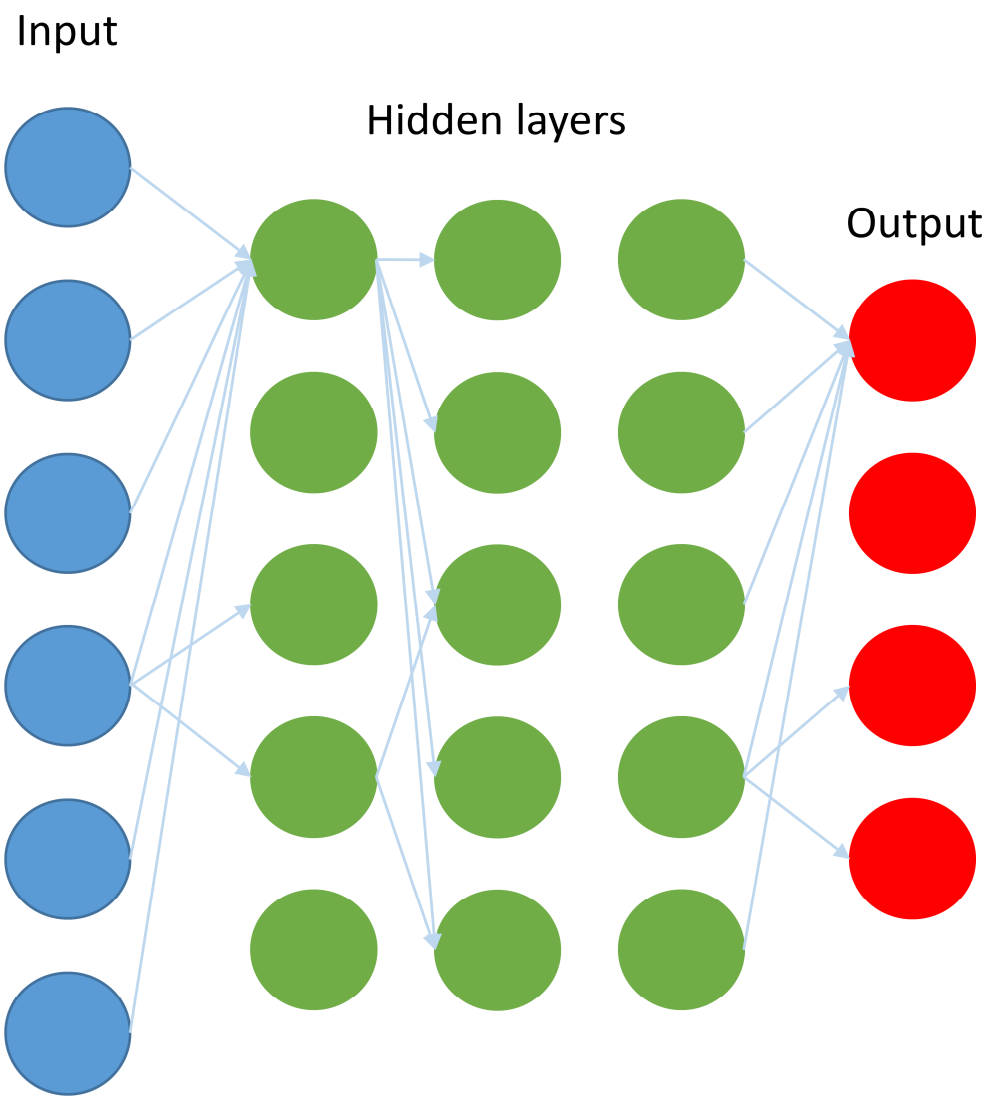


Figure 1.7. Schematic structure of an Artificial Neural Network (ANN). Only some connections are shown. Every neuron has connections to every neuron in the next layer

Activation function

Every neuron takes its inputs and computes their linear combination with the weight of the neuron. This procedure is linear; however, a nonlinear function that is applied to the outcome of this computation, activation function. The simplest activation function is a sigmoid (logistic) function which has already been described.

Another interesting activation function is called Rectified Linear Unit (ReLU). It is a linear function on the positive half of the X axis and it is always zero on the negative side. The idea to use this type of activation was inspired by real neurons in the brain. A real neuron does not respond to the most inputs and most of the time does not perform any action. This leads to sparsity of neural networks (Glorot et al., 2011).

Optimization

Machine learning algorithms need to be trained. Training is the process of finding optimal parameters of an algorithm. One can think of it as optimization of a loss function in the parameter space. Optimization algorithms look for the set of parameters, which minimize the cost function. In many cases it is not possible to find the global minimum. For example there is no method up to date to find global minima for ANN.

The most widely known optimization algorithm is the gradient descent algorithm (GD) (Cauchy, 1847).

It takes tremendous amount of resources to compute the gradient on the whole dataset. The gradient can be estimated by using only small random sample of the data. Such algorithm is called Stochastic Gradient Descent (SGD) (Robbins and Monro, 1951, Kiefer and Wolfowitz, 1952).

Vanishing gradient problem

Weights of the neural network are being updated according to the gradients computed out of errors. If neural network contains a large number of layers, these gradients may become very small and the training may stop (Hochreiter, 1991).

A possible solution to this problem is the introduction of residual connections. In this case so-called skip-connections jump over a layer. Networks with such connections are called Residual Networks (He et al., 2016). Residual networks may contain hundreds of layers. Residual connections allow training of very deep networks with a large number of layers.

Gradients can not only become too small. They can also become too large, a phenomenon called 'exploding gradients'. One can overcome this problem with for example gradient clipping, the maximal absolute value of the gradient is limited to the certain cut off value (Pascanu et al., 2012, Bengio et al., 2013).

Dropout

Neural networks usually have large amount of parameters. As a result, they can overfit easily. One of the most efficient ways to prevent overfitting is to randomly switch off some neurons on every training iteration. It lets the network learn how to predict classes when some signals are not available. Therefore neuronal activations become sparse. This method is called dropout and it is highly efficient (Hinton et al., 2012).

1.8.9 Learning the temporal structure

The algorithms described above classify every data point with irrespective of the events that occurred in the past. This is true regarding

Random Forest, Neural Networks, logistic regression and many other methods. However, it is well known that scoring of sleep stages is dependent on the past. Therefore, taking local temporal information into account is important for automatic sleep scoring.

However, when the human expert scores sleep they should not rely on the position of the epoch in the recording. But an expert can take into account information about several previous epochs to decide on the stage of the current epoch. Learning very long patterns e.g. sleep cycles might damage the performance when the sleep structure is not normal, e.g. if sleep is disrupted or is not continuous in a recording (multiple sleep latency test (MSLT) recorded continuously over 9 h leading to long intervals of wakefulness in-between the tests). Sleep structure is severely affected in obstructive sleep apnea patients, in patients with narcolepsy and elderly people with sleep difficulties. If we learn long sequences of healthy subjects, it might bias the algorithm and such models would perform poorly on the recordings of altered sleep. Algorithms trained on recordings of healthy sleep might also not be optimal for detection of sleep onset REM sleep episodes (SOREM sleep), which is important e.g. in the screening of narcolepsy. Therefore, we decided to limit the length of the sequence available to an algorithm while scoring.

Hidden Markov Model (Stratonovich, 1960)

Let's consider popular example of the HMM applied to the weather prediction (Rabiner, 1989, Resch, 2004). Imagine that you have a friend who lives in another city. The two of you are playing a game. You do not know how the weather is in the city is, but your friend tells you what they do every day and your friend's activity is dependent on the weather. Your task in this game is to find out what was the weather in the place of your friend every day.

Let assume there are two possible types of weather: sunny and rainy. Those are our hidden states. They are called hidden because you cannot observe them directly. And the weather tomorrow is dependent on the weather of today. Transition into another weather condition happens with a certain probability. If it was rainy today, then there is a 60 % chance that it will also rain tomorrow, and there is a 40 % chance that it will be sunny. If today is sunny, then there is a 30 % chance that it will be rainy tomorrow, and there is a 70 % chance that it will be sunny tomorrow too. These probabilities are called transition probabilities. You also know the probabilities of sunny and rainy weather on the first day of your game (initial probabilities), you know that your friend is more likely to go biking or meet with their friends for a coffee on a sunny day, and that he is more likely to stay at home to do chores on a rainy day. You know the exact probabilities of each activity depending on the weather condition. These probabilities are called emission probabilities as the system 'emits' an observation.

The problem of the estimation of parameters of HMM given a set of observations cannot be solved exactly. However, the local optimum of maximum likelihood estimation can be found using the Baum-Welch algorithm (Welch, 2003). This is the learning step. To predict a new sequence of hidden states, we need the parameters of the HMM and new observations. The algorithm to infer most probable sequence of hidden states is called the Viterbi algorithm (Viterbi, 1967).

A great advantage of a HMM is its simplicity, whereas its disadvantage is the short memory which is one step only.

Recurrent Neural Networks (RNNs)

Another way to learn temporal information is to use Recurrent Neural Networks (RNNs). RNNs are similar to a common neural network, but a neuron receives not only the activation of the previous layer on the current time step, but also its own activation from the previous time step. This way the network can "see" its past. If we unroll the network in time direction, we will see that it has large amount of layers in this direction. It is therefore not surprising that RNNs suffer from the vanishing gradient problem.

A special type of neurons was developed by Hochreiter et al. (Hochreiter and Schmidhuber, 1997). This network is called Long-Short Term Memory (LSTM) network. It contains a set of gates which prevent a gradient from becoming too small. It is also possible to let the network "see" not only the past but also future. In this case the network is called bidirectional. In this case, to predict something, the full sequence needs to be available and it is not possible to predict data online.

1.8.10 Unsupervised learning

Unsupervised learning deals with unlabeled data. For the most classification problems supervised learning is usually used in case good labeled datasets are available. Sometimes this is not the case. If the labeled data is not available in required amount, one can use unsupervised learning.

We have already considered one of the simplest unsupervised machine learning algorithms – K-means. There are plenty of other clustering algorithms (Xu and Wunsch, 2005). Unfortunately clustering has limited performance with complex data. Many problems have been successfully addressed with unsupervised learning through ANNs in image analysis (Le, 2013, Oord et al., 2016) , and natural language processing (NLP) (Mikolov et al., 2010, Conneau

et al., 2017, Salakhutdinov and Hinton, 2009, Artetxe et al., 2017, Lample et al., 2017).

Artificial Neural Network (ANN) based methods for unsupervised learning have been developing rapidly. The idea of the autoencoder (Hinton and Salakhutdinov, 2006) is to train neural network to reproduce its own input. Schematic autoencoder is shown on the Figure 1.8. But there is a constraint, the layer in the middle of the network has low number of parameters, the so-called 'bottleneck'. The bottleneck layer is considered to contain an internal representation or code. The low number of parameters in the bottleneck enforces internal representation with low dimensionality. The part of the network before the bottleneck is called the encoder, the part thereafter the decoder. The encoder encodes the input into the internal representation and the decoder decodes the signal from the internal representation.

The dimensionality of the internal representation can be further reduced to 2 or 3 and thus the data can be represented in a plane, which is helpful for visualization. The most advanced algorithm to do so is t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008). t-SNE is also a dimensionality reduction algorithm, but it is not an autoencoder. t-SNE tries to keep the distance between the data points the same as in the original data. This is the reason why it is great for data visualization.

After the training, the internal representation can be used for clustering. It is also possible to use the decoder and the representation it has learned as a pretrained network for further supervised learning (Erhan et al., 2010). This approach is useful when very small amount of labeled data are available.

Denoising autoencoders (DAE) (Vincent et al., 2008) can be used for data denoising. The idea is to train the network to reconstruct the original signal from the same signal with added noise. Moreover, adding noise to the data

works as a regularization and prevents the autoencoder from overfitting. Another regularization method is to enforce sparsity of the internal representation (Andrew, 2011), such a network is called Sparse Autoencoder (SAE).

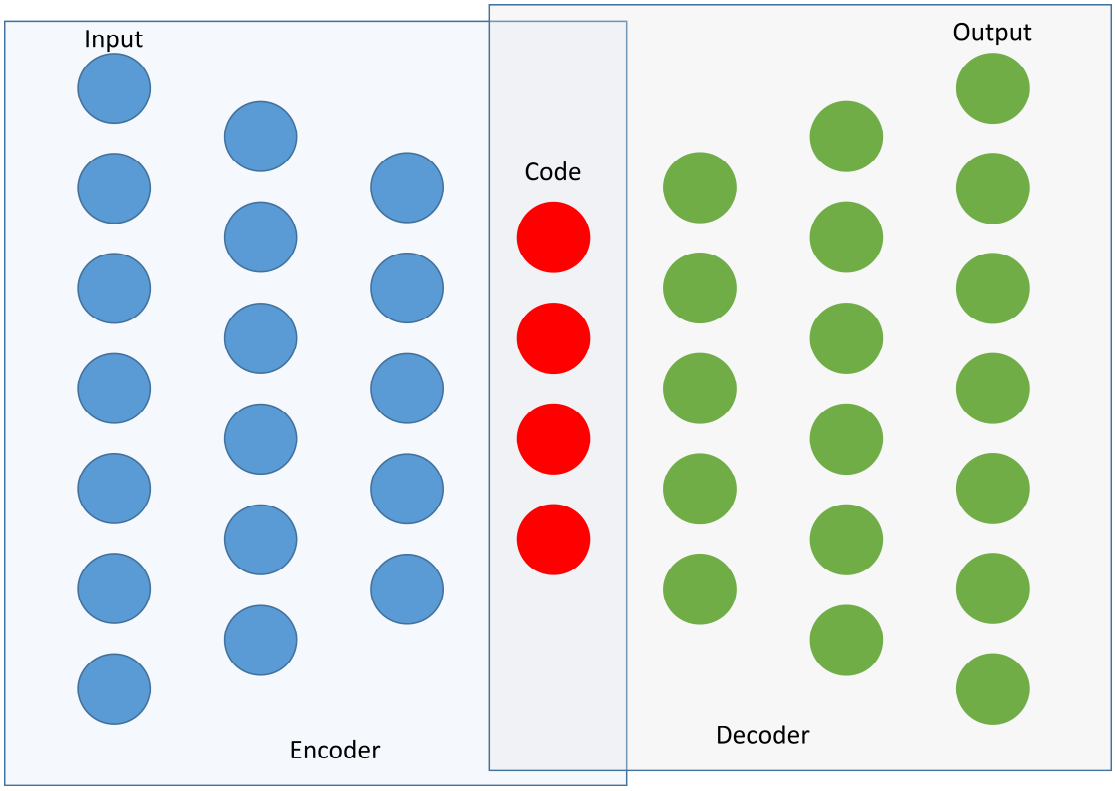


Figure 1.8. Schematic structure of an autoencoder. Left part illustrates an encoder, right part a decoder. The 'Bottleneck' is in the middle, it is the layer which contains the vector of internal representation (Code). Every neuron has connections to every neuron in the next layer

The most recent and advanced autoencoder is the Variational Autoencoder (VAE) (Kingma and Welling, 2013). This type of autoencoder learns the properties of the distribution of the internal state, or the latent variable model (Everett, 2013). It means that we can sample points from this distribution and decode them into the desired signal. It may also be used to generate new data.

1.8.11 Performance evaluation

Hypnogram

As mentioned before, hypnograms provide a good overview over the structure of sleep and it is possible to see the difference between the scoring performance of different algorithms quickly by just looking at the hypnograms. Still, in order to choose the best method, it is necessary to have a numerical quantification of the comparison.

To begin with, let us consider a binary classification, i.e. we have only two types of labels: positive and negative as it is the case for artifact detection. We label every epoch as either contaminated or clean. This is also the case with medical tests in which we have two possible outcomes: a person is either ill or healthy.

Accuracy

The simplest measure of the performance such a classifiers is the percentage of correctly identified classes, accuracy. Unfortunately, though, it does not work well if we have an unbalanced distribution of the classes. Let us imagine we need to test patients for a rare disease. Our test might always give a negative result. Obviously, such a test does not make sense at all, even though the accuracy would be close to 100%.

Type I and II errors

Let us look at the rare disease testing in more detail. Imagine that the person is healthy but the outcome of the test for the disease would be positive. This type of outcome is called a false positive outcome or type I error. If the person is ill and the outcome of the test is negative, then there is a false negative outcome or type II error.

Sensitivity and Specificity

Most widespread performance measures of binary classifiers are sensitivity and specificity (Altman and Bland, 1994). Sensitivity is a ratio of correctly identified positive examples (true positive, TP) to the total number of positive examples (P). It is also called recall or true positive rate (TPR) or probability of detection. Another relevant measure is precision. Precision is the percentage of true positive outcomes among all data classified as positive. Specificity (SPC) or false negative rate (FNR) is the percentage of incorrectly identified positive examples (false positive, FP) among all negative examples (N):

$$TPR = TP/P \tag{1.2}$$

$$SPC = TN/N \tag{1.3}$$

ROC curve

Every binary classifier has a certain threshold which separates the two classes. Adjusting this threshold can make a classifier more sensitive or more specific; there is always a trade-off between errors of type I and II. Receiver operating characteristic (ROC) (Green and Swets, 1966) were introduced to see a bigger picture. If we vary the threshold and measure true positive rate and false positive rate for every threshold and plot them for every threshold, we will get a curve, the ROC curve. Generally, a ROC curve of a better classifier has a larger area under the curve. The area under the curve can vary from 0.5 for completely random to 1 for an ideal classification.

F1 score

As we have already seen in classification problems with highly imbalanced classes we need both sensitivity and specificity to understand if our classifier is good. This might be inconvenient. Fortunately, we can compute

a single measure of the classification quality. One of such measures is the F1-score (Sørensen, 1948, Dice, 1945). It is a harmonic mean of recall and precision.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{1.4}$$

1.8.12 Validation

As previously mentioned, one of the most important things to look at is overfitting. A good way to control for it is to split data into a training and validation part (Arlot and Celisse, 2010). This method is also called hold-out. After the split we can train our model on the training part and then compute performance metrics on the validation part.

Our results on the validation part are meant be close to the real life results in case our data set was representative. In some cases, we have certain hyperparameters (e.g. parameters of regularization, number of trees in the RF, number of layers in a network) to tune, or we need to choose the best model. Then, we split our data into three parts: training, validation and test. We can use validation part to choose the best model and then evaluate its performance using test set. A rule of thumb for the split is to put 70 % of the data into the training set, 15 % to the validation set and 15 % to the test set. It is needed to be done due to the following point. If we compare different models, we have a chance to overfit on this level which is similar to multiple hypotheses testing problem, therefore to avoid this validation is needed.

Leave-one-out method (Arlot and Celisse, 2010, Stone, 1974)

If we have a small amount of data only, we may train on all our data (except one example) and test on this one example only. Of course, reliability of such a solution will be compromised. Thus, we might iterate through the

data each time leaving out another example. With the average of obtained performance estimates, we will achieve a reliable result. The disadvantage of leave one out method is that we have to train N models, where N is the number of examples. This might not be feasible, especially for complex neural networks.

K-fold cross validation (Arlot and Celisse, 2010)

This approach is similar to the leave-one-out method. We should split the dataset into k subsets. Then train the model k times, each time using one of the subset for testing and the rest of the data for training.

1.9 Automatic sleep scoring

Martin et al. (Martin et al., 1972) classified sleep (EEG and EOG signals) with the help of a decision tree. Authors reported that computer performed 7% worse than human experts. Unfortunately, the number of participants in the study was insufficient to judge if computer can compete with a human expert. A similar method was developed by Louis et al. (Louis et al., 2004).

Stanus et al. (Stanus et al., 1987) developed and validated two automatic sleep classification algorithms: the first one was built upon autoregressive model; the second algorithm employed power of specific frequency bands and Bayesian decision theory. In addition to EEG, both algorithms used 2 EOG signals in order to detect eye movements and an EMG signal to take muscle tone into account. It was reported in the paper that there was 80% agreement with an expert.

Power density spectra of the EEG are useful not only for sleep scoring. Fell et al. (Fell et al., 1996) developed automatic scoring methods with non-linear features (correlation dimension, Kolmogorov entropy, Lyapunov

exponent). This study has shown that these parameters increase overall performance. Park et al. (Park et al., 2000) developed a rule-based algorithm and reported high performance. Authors reported that their method performed well also on the patient data.

One of the commercially successful products of automatic sleep analysis is the SIESTA project (Klosh et al., 2001). The software is called Somnolyzer 24x7. The program performs quality check of the data using histograms. The classification is performed by a decision tree using extracted features. Input data include one EEG channel, two EOG channels and one EMG channel. Somnolyzer 24x7 also adjusts borders of REM episodes taking into account the scoring rules. The 3-min rule for intrusions of stage 1 into stage 2 was also applied by a correction procedure (Anderer et al., 2005). Somnolyzer 24x7 was validated on a big database. The database was comprised of the data of 90 patients with various sleep disorders and approximately 200 control subjects. Recordings were scored by several experts. Somnolyzer 24x7 showed high agreement with the result of expert scoring (Anderer et al., 2005). Authors reported that for some recordings the use of EMG was suboptimal. In such cases the software substituted the EMG signal with the content of high frequency range of the EEG and EOG signals. It increased the agreement of the algorithm and the experts in the study. In our study, we also noticed that anomalies in the EMG signal are detrimental for automatic scoring.

Since Artificial Neural Networks (ANN) have been shown to be good for pattern recognition (Bishop, 2016, Goodfellow et al., 2016), a number of attempts to classify sleep using ANNs have been made. Schaltenbrand et al. (Schaltenbrand et al., 1993) used ANN for sleep scoring with an input comprised of 17 features. These features were extracted from PSG signals.

Reported accuracy was close to 90 %. Längkvist (Längkvist et al., 2012) classified sleep stages using Restricted Bolzmann machines.

Convolutional Neural Networks (CNNs) are meant to solve visual pattern recognition tasks. Since sleep scoring is such a task, it is very natural to apply a CNNs to score sleep. Tsinalis (Tsinalis et al., 2016) applied a CNN to the sleep recordings (raw EEG). CNNs are very efficient in learning complex patterns and they are meant to interpret visual data in a similar way as a brain (Fukushima and Miyake, 1982). The disadvantage of this approach is that analyzing raw data requires much more computational resources and data for training.

It is also possible to score sleep in an unsupervised way; such attempts have been made for both human (Agarwal and Gotman, 2001, Grube et al., 2002, Gath and Geva, 1989) and animal sleep (Sunagawa et al., 2013, Libourel et al., 2015).

2. Automatic artifact detection in single channel sleep EEG recordings

Alexander Malafeev[1,2], Ximena Omlin[2,3], Aleksandra Wierzbicka[4], Adam Wichniak[5], Wojciech Jernajczyk[4], Robert Riener[3]and Peter Achermann[1,2,6]

[1] Institute of Pharmacology and Toxicology, Chronobiology and Sleep Research, University of Zurich, Zurich, Zurich, Switzerland

[2] Neuroscience Center Zurich, University of Zurich and ETH Zurich, Zurich, Switzerland

[3] Sensory-Motor Systems Lab, ETH Zurich, Zurich, Switzerland

[4] Sleep Disorders Center, Department of Clinical Neurophysiology, Institute of Psychiatry and Neurology in Warsaw, Warsaw, Poland

[5] Third Department of Psychiatry and Sleep Disorders Center, Institute of Psychiatry and Neurology in Warsaw, Warsaw, Poland

[6] Zurich Center for Interdisciplinary Sleep Research, University of Zurich, Zurich, Zurich, Switzerland

Disclosure: The authors declare no conflicts of interest.

Author contributions: AM and PA designed the analyses; AM conducted the analyses; XO, RR, and PA; AW, AW, and WJ collected the data; AM and PA wrote the manuscript and all authors commented and accepted the final version.

Accepted for publication in the Journal of Sleep Research.

2.1 Abstract

Quantitative EEG analysis (e.g. spectral analysis) has become important tool in sleep research and sleep medicine. However, reliable results are only obtained if artifacts are removed or excluded. Artifact detection is often performed manually during sleep stage scoring, which is time consuming and prevents application to large data sets. We aimed to test performance of mostly simple algorithms of artifact detection in polysomnographic recordings, derive optimal parameters and test their generalization capacity.

We implemented 14 different artifact detection methods, optimized parameters for derivation C3A2 using receiver operator characteristic curves of 32 recordings and validated them on 21 recordings of healthy participants and 10 recordings of patients (different laboratory) and consider the methods as generalizable. We also compared average power density spectra with artifacts excluded based on algorithms and expert scoring. Analyses were performed retrospectively.

We could reliably identify artifact contaminated epochs in sleep EEG recordings of two laboratories (healthy participants and patients) reaching good sensitivity (specificity 0.9) with most algorithms. The best performance was obtained using fixed thresholds of the EEG slope, high frequency power (25-90 Hz or 45-90 Hz) and residuals of adaptive autoregressive models.

Artifacts in EEG data can be reliably excluded by simple algorithms with good performance and average EEG power density spectra with artifacts exclusion based on algorithms and manual scoring are very similar in the frequency range relevant for most applications in sleep research and sleep medicine allowing application to large data sets as needed to address questions related to genetics, epidemiology, or precision medicine.

Keywords: Computerized Analysis, Computational Neuroscience, EEG Spectral Analysis, MSLT

2.2 Introduction

Electroencephalographic (EEG) recordings may contain artifacts from many different sources, which is detrimental for quantitative EEG analysis. Thus, artifact detection and exclusion are essential for quantitative EEG analysis. In sleep research, manual marking of artifacts during sleep stage scoring is common which is time consuming and prevents application to large data sets, i.e. as needed in genetics, epidemiology, or precision medicine. Thus, automated methods revealing consistent results are needed. Here we focus on simple approaches applicable to a single EEG derivation as they should be easily implementable in small portable devices or work on-line and without prior sleep stage scoring.

Technical artifacts, for example power line noise, may be removed by a band-stop filter (notch filter). However, biological artifacts like muscle activity, movement, and ocular artifacts and electrical activity of the heart are more difficult to detect as they have a broad variation of appearance.

In general, we have to dissociate between artifact detection (and exclusion for quantitative analyses) and artifact removal ("subtraction" from the EEG). It is difficult to solve artifact subtraction problem exactly. Some signal from the artifact source may remain and part of the useful signal can be removed. It also often requires multiple channels (Winkler et al., 2011, Delorme and Makeig, 2004), which are not necessarily available with portable devices.

Ocular artifacts can be removed by number of techniques, for example regression analysis (Semlitsch et al., 1986), blind source separation (BSS), or

independent component analysis (ICA) (Comon, 1994, Girolami, 1998, Lee et al., 1999, Gavelin et al., 2004, Groppe et al., 2009).

Muscle and movement artifacts can tremendously affect the spectra of the EEG recordings, especially in the higher frequency range. These types of artifacts are difficult to detect since they are very variable. However, muscle artifacts have some characteristic properties. Most of the spectral power of a muscle contraction event in the EEG is above 25 Hz (Gotman et al., 1981) and muscle artifacts contaminate the high frequency range (20-80 Hz) with the peak around 40 Hz and also affect lower frequencies (Goncharova et al., 2003). Since muscle artifacts contaminate the higher frequency range it is possible to apply a low-pass filter (e.g. Gevins et al., 1975). However, this may not be the best approach if EEG components above 20-25 Hz are of interest. One of the approaches often applied to avoid problems of filtering or artifact subtraction is the rejection of segments with artifacts. We used this approach and identified 20-s or 30-s EEG segments with artifacts.

We implemented 12 algorithms previously published and developed two new ones (Table 2.1). Many older papers on artifact detection did not report the performance of the algorithms. We estimated the optimal parameters of the algorithms and evaluated their performance on two types of recordings: nocturnal sleep of healthy participants and patients and a mixture of sleep and wakefulness in a multiple sleep latency test (MSLT) recorded continuously over approximately 9 h in patients. Parameter estimation and validation was performed on independent datasets.

2.3 Materials and methods

2.3.1 Data sets

We analyzed two data sets:

1) Polysomnographic (PSG) recordings of an experiment with vestibular stimulation (Omlin et al., 2018).Three nights (8 hours) of 18 healthy young males (age: 20-28 years; mean: 23.7 years) were recorded: two motion nights (rocking until sleep onset; rocking for first 2 hours after lights out), and a baseline without motion. Recordings included 12 EEG channels, placed according 10-20 system, 2 EOG channels, 1 chin EMG, 1 ECG channel and respiration (chest and abdomen). Data were recorded with the polygraphic amplifier Artisan (Micromed, Mogliano, Veneto, Italy). The signals were sampled at 256 Hz (Rembrandt DataLab; Version 8.0; Embla Systems, Broom Field, CO, USA). Analogue signals were filtered with a high pass filter (EEG: -3 dB at 0.16 Hz; EMG: 10 Hz; ECG: 1 Hz) and an anti-aliasing low-pass filter (-3 dB at 67.4 Hz). Sleep stages (20-s epochs) were scored according to standardized criteria (Iber et al., 2007). Recordings were performed in the sleep laboratory of the Institute of Pharmacology and Toxicology at the University of Zurich. The Institutional Review Board of the Swiss Federal Institute of Technology in Zurich (ETH Zurich) approved the study. In total, this dataset comprised 53 PSG nighttime recordings of healthy participants.

2) PSG data recorded in patients with hypersomnia (2 subjects) and narcolepsy (3 subjects) who underwent a multiple sleep latency test (MSLT). The EEG was recorded continuously for approximately 9 h throughout the MSLT. In addition, a night of sleep was recorded in each patient. PSG included 6 EEG, 2 EMG, 2 EOG channels and 1 ECG. Data were sampled at 200 Hz (polygraphic amplifier Grass Technologies AURA PSG). Analogue signals were filtered with a high pass filter (EEG: -3 dB at 0.5 Hz) and an anti-aliasing low-

pass filter (-3 dB at 50 Hz). Sleep stages (30-s epochs) were scored according to standardized criteria (Rechtschaffen and Kales, 1968). PSG recordings were performed at the Sleep Disorders Center, Department of Clinical Neurophysiology, Institute of Psychiatry and Neurology in Warsaw, Warsaw, Poland. The Institutional Review Board of Institute of Psychiatry and Neurology approved the study. In total, this dataset comprises 5 sleep and 5 MSLT recordings of narcoleptic (n=3) and hypersomnia patients (n=2).

To illustrate the sleep structure in the EEG and the occurrence of large artifacts spectrograms were calculated. Power density spectra were determined for 20-s or 30-s epochs (FFT; average of five 4-s or six 5-s epochs without overlap; Hanning window). Spectra are plotted and color-coded on a logarithmic scale (spectrograms, Figures 2.2 to 2.4).

Artifacts were visually scored by experts in both datasets on an epoch basis (i.e. each 20-s or 30-s epoch has a label whether it contains an artifact or not). Please note that in dataset 1 (healthy subjects), only severe artifacts were visually identified. Scorers were instructed to mark only severe artifacts. Afterwards a semiautomatic procedure based on EEG power in the 20-40 and 0.75-4.5 Hz range was applied to detect small artifacts. Artifact markings of the semiautomatic procedure were used only in original study (Omlin et al., 2018). Artifacts in the second dataset (patients) were scored by the first author and all artifacts were marked. This might explain that the FPRs in the second dataset were smaller than in the test data of the first dataset (Table 2.2).

We analyzed derivation C3A2 in the context of this paper. The parameters of the algorithms are to some degree dependent on the derivation used. If referential (mastoid reference) derivations (frontal, central, occipital) as classical for sleep recordings are used we do not expect that adaptations are

needed. However, working with e.g. bipolar recordings would require adaptation of the parameters.

Method	Resolution	Online mode possible?
Amplitude thresholding, fixed threshold (ATf)	Sample	Yes
Amplitude thresholding, statistical threshold (ATs)	Sample	No
Slope thresholding, fixed threshold (STf)	Sample	Yes
Slope thresholding, statistical threshold (STs)	Sample	No
Zero Crossings (ZC)	Sample	Yes
Mean Crossings (MC)	Sample	Yes
Power thresholding 25-90 Hz (PT25)	Sample	Yes
Power thresholding 45-90 Hz (PT45)	Sample	Yes
Power thresholding (average power of epoch) (PTe)	Epoch	Yes
Autoregressive (AR) model	Sample	No
Adaptive AR model, fixed threshold (aARf)	Sample	Yes
Adaptive AR model, statistical threshold (aARs)	Sample	No
K-means (KM) clustering	Epoch	No
Hidden Markov Model (HMM)	Epoch	No

Table 2.1. Overview of the applied algorithms and their abbreviations used. Most of the algorithms return whether a single sample belongs to an artifact or not (resolution sample), but some return whether a whole epoch (20 or 30 s) contains an artifact or not (resolution epoch). It is also indicated whether an algorithm could be implemented on-line (yes) or whether the entire recording is needed first (no). KM and HMM are the two newly developed algorithms. The algorithms are detailed in Supporting Information.

We mainly focused on simple algorithms that are easy to implement and were used in the past. Two additional algorithms were developed and tested. In contrast to the other algorithms, these two have no tunable parameters as they cluster the data in two categories (no artifacts, artifacts; see Supporting Information).

Most of these algorithms produce a classification for each sample (Table 2.1), i.e. an outcome of an algorithm is an array with labels for each sample

whether it belongs to an artifact or not. We translated this information into an epoch wise classification in order to be able to compare the outcome of an algorithm with our expert classification. We classified an epoch as an artifact if it contained at least one sample identified as an artifact.

2.3.2 Algorithms

The implemented algorithms (Table 2.1; abbreviations) and the corresponding parameters (derived and applied are listed in Table 2.2) are described in the Supporting Information.

We do not expect a noticeable influence of the sampling rates and filter settings of the recording equipment used because the algorithms were chosen to work in the frequency range of 0.5-90 Hz. This is evident as parameters derived on healthy participants were transferable to patients recorded with a different system. However, for sampling rates < 200 Hz adaptations would be needed.

2.3.3 Evaluation of the performance of the algorithms

To evaluate the performance of the algorithms, we randomly split the data of the first dataset (sleep data) into a training and testing set in proportion of 60 to 40 percent (32 and 21 recordings). We computed receiver operator characteristic (ROC) curves (Green and Swets, 1966) for each algorithm and recording of the training dataset. To compute the ROC curves parameters of the algorithms (thresholds) were systematically varied in a certain range (Table 2.2). ROC curves are plots which give an understanding about the performance of a binary classifier. On the x-axis, the false positive rate (FPR – percentage of clean epochs marked as containing artifacts) and on the y-axis the true positive rate (TPR – percentage of epochs with artifacts

which were marked as having artifacts) are plotted. One varies the threshold and calculates FPR and TPR for each value of a threshold. Plotting of these points forms the ROC curve. In case of random results, the ROC curve would be a straight line with the area under the curve (AUC) equal to 0.5. The AUC is a marker of the quality of an algorithm, the larger the AUC, the better the performance of the algorithm. ROC curves for ATf, STf and PT25 are illustrated in Figure 2.1.

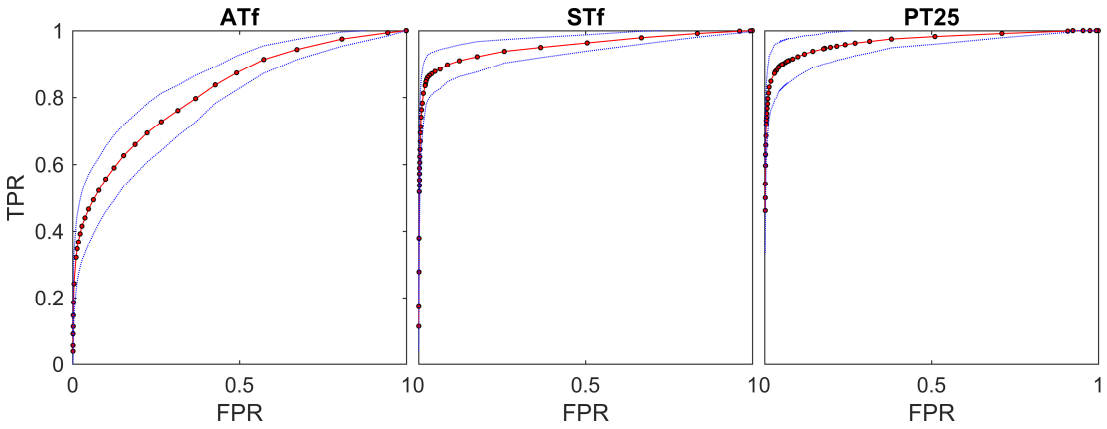


Figure 2.1. Average ROC curves (mean and standard deviation across training set) for the algorithms "Amplitude Thresholding, fixed threshold ATf" (left), "Slope thresholding, fixed threshold STf" (middle) and "Power thresholding 25-90 Hz PT25" (right). Dots with the red line show average ROC curve among recordings in the training set. TPR: true positive rate; FPR: false positive rate. Blue curves depict standard deviations

There is not a single way to choose an optimal threshold for the application of the algorithms as sensitivity and specificity cannot be increased concomitantly (Habibzadeh et al., 2016). Moreover, it is application dependent whether priority is given to sensitivity or specificity. In some applications the "costs" of false negatives are large, in other applications the "costs" of false positives. In our case, false negatives may distort average spectra in the frequency range of interest, whereas exclusion of some

additional clean epochs (false positives) would not affect average spectra. After visual inspection of the ROC curves we decided to apply a threshold corresponding to a false positive rate close to 0.1 (not based on optimization). Our choice of a FPR fixed at 0.1 has consequences, i.e. it leads to the rejection of 10 % of all epochs not marked as artefactual by the expert to be rejected. As however, experts only marked severe artifacts (dataset 1, see above) we consider this as justifiable. Experts excluded 7 ± 3 % of epochs. Therefore, the total rejected epochs should theoretically fluctuate around 16.3 %.

For the test data sets, we applied those thresholds and computed specificity (percentage of artifact free epochs correctly marked as artifact free) and sensitivity (equal to TPR) and compared resulting performance measures with performance on the training set and average power density spectra of NREM sleep were calculated to assess the impact of artifact exclusion. Additionally, we applied the algorithms with the same thresholds to patient data of the second data set and validated performance additionally on sleep and MSLT recordings.

2.4 Results

2.4.1 Derivation of parameters (thresholds) of the algorithms

Areas under the ROC curves, optimal thresholds (we chose them in a way that FPR ~ 0.1), and TPR resulting from the training data sets are depicted in Table 2.2 (columns 2 to 4). Seven algorithms showed quite good performance (AUC > 0.95) with PT25 showing the best performance, i.e. largest AUC and TPR.

	Dataset 1 (training data)			Dataset 1 (test data)	
	AUC (std)	**Thres**	**TPR (std)**	**TPR (std)**	**FPR (std)**
Amplitude thresholding, fixed threshold (ATf)	0.820 (0.059)	168.333 μV	0.556 (0.097)	0.496 (0.122)	0.097 (0.062)
Amplitude thresholding, statistical threshold (ATs)	0.819 (0.059)	5.292 σ	0.489 (0.181)	0.420 (0.151)	0.087 (0.075)
Slope thresholding, fixed threshold (STf)	0.953 (0.023)	928336.64 μV/s	0.905 (0.051)	0.886 (0.067)	0.103 (0.067)
Slope thresholding, statistical threshold (STs)	0.952 (0.023)	3.700 σ	0.905 (0.041)	0.868 (0.137)	0.138 (0.093)
Zero Crossings (ZC)	0.866 (0.068)	36.250 #/s	0.692 (0.126)	0.644 (0.105)	0.072 (0.041)
Mean Crossings (MC)	0.917 (0.047)	40.000 #/s	0.791 (0.100)	0.756 (0.074)	0.079 (0.042)
Power thresholding 25-90 Hz (PT25)	0.966 (0.028)	8.400 μV^2	0.923 (0.058)	0.905 (0.079)	0.108 (0.078)
Power thresholding 45-90 Hz (PT45)	0.962 (0.031)	3.143 μV^2	0.909 (0.073)	0.899 (0.083)	0.097 (0.069)
Power thresholding (PTe)	0.926 (0.037)	5.263 μV^2	0.780 (0.073)	0.749 (0.075)	0.091 (0.083)
Autoregressive Model (AR)	0.954 (0.023)	3.458 σ	0.911 (0.046)	0.878 (0.121)	0.137 (0.084)
Adaptive AR, fixed threshold (aARf)	0.956 (0.021)	11.111 μV	0.897 (0.065)	0.887 (0.071)	0.111 (0.080)
Adaptive AR, statistical threshold (aARs)	0.956 (0.021)	3.333 σ	0.906 (0.050)	0.884 (0.113)	0.138 (0.074)
K-Means (KM)				0.406 (0.151)	0.190 (0.118)
HMM				0.652 (0.176)	0.269 (0.135)

Table 2.2. Parameters and performance of the algorithms. The first three columns correspond to the training of the algorithms. Area under the curve (AUC) with its standard deviation, optimal threshold (Thres; σ, standard deviation), true positive rate (TPR) at a false positive rate (FPR) close to 0.1.

	Dataset 2 (MSLT data)		Dataset 2 (sleep data)	
	TPR (std)	**FPR (std)**	**TPR (std)**	**FPR (std)**
Amplitude thresholding, fixed threshold (ATf)	0.514(0.262)	0.061 (0.103)	0.405 (0.128)	0.014 (0.024)
Amplitude thresholding, statistical threshold (ATs)	0.310 (0.159)	0.026 (0.033)	0.398 (0.153)	0.014 (0.023)
Slope thresholding, fixed threshold (STf)	0.946 (0.035)	0.143 (0.100)	0.696 (0.140)	0.043 (0.052)
Slope thresholding, statistical threshold STs)	0.763 (0.136)	0.076 (0.086)	0.726 (0.232)	0.055 (0.052)
Zero Crossings (ZC)	0.949 (0.045)	0.338 (0.071)	0.727 (0.109)	0.169 (0.263)
Mean Crossings(MC)	0.951 (0.047)	0.273 (0.084)	0.720 (0.100)	0.131 (0.236)
Power thresholding 25-90 Hz (PT25)	0.994 (0.002)	0.245 (0.107)	0.902 (0.072)	0.076 (0.075)
Power thresholding 45-90 Hz (PT45)	0.988 (0.005)	0.206 (0.093)	0.855 (0.094)	0.045 (0.052)
Power thresholding (PTe)	0.970 (0.028)	0.228 (0.125)	0.833 (0.017)	0.059 (0.069)
Autoregressive Model (AR)	0.865 (0.104)	0.109 (0.111)	0.820 (0.252)	0.176 (0.166)
Adaptive AR, fixed threshold (aARf)	0.990 (0.004)	0.326 (0.123)	0.922 (0.052)	0.134 (0.102)
Adaptive AR, statistical threshold (aARs)	0.881 (0.084)	0.111 (0.111)	0.823 (0.256)	0.121 (0.090)
K-Means (KM)	0.902 (0.077)	0.183 (0.088)	0.346 (0.170)	0.050 (0.041)
HMM	0.924 (0.025)	0.216 (0.083)	0.754 (0.179)	0.368 (0.185)

Columns 4 and 5 represent the TPR and FPR obtained by applying the algorithms to the MSLT and sleep data of dataset 2 (patients). K-means (KM) clustering and hidden Markov models (HMM) did not work on sleep data of dataset 1 and 2 which contained a very small number of epochs with artifacts which is insufficient that these unsupervised classifiers could learn that artifacts are a separate class.

2.4.2 Testing of performance on independent data sets

The performance of the algorithms was tested on the test data of dataset 1 and data (MSLT and sleep) of dataset 2 applying the derived thresholds.

Performance of the algorithms (Table 2.2, columns 5-6), i.e. the TPR was somewhat lower for the test data than for the training data. PT25 was again performing best. Performance of algorithms KM and HMM was not satisfactory (Table 2.2). Examples of artifact detection applied to single sleep recordings of dataset 1 are illustrated in Figures 2.2 and 2.3. Note that performance of some algorithms varies considerably from recording to recording (e.g. STs or AR in Figure 2.2 and 2.3). The values of TPR and FPR reported in the Table 2.2 are average ones. The fluctuations in the performance of the algorithms applied to different recordings are reflected in the standard deviations shown in brackets.

Performance of the algorithms applied to patient data of a different laboratory (dataset 2) was also good (True Positive Rate (TPR) ≥ 0.9; False Positive Rate (FPR) ~ 0.1; Table 2.2, columns 7-10; sensitivity = TPR; specificity = 1 - FPR), thus, the determined thresholds are generalizable. KM and HMM performed well on the MSLT data with a lot of intermittent wakefulness (Table 2.2, columns 7-8). However, performance on sleep data (Table 2.2, columns 9-10) was not satisfactory. Figure 2.4 illustrates artifact detection in a MSLT recording of dataset 2.

2.4.3 Effect of artifact exclusion on NREM sleep power density spectra

An important purpose of artifact identification is to be able to obtain clean average power density spectra for further evaluation. Figure 2.5 (top

rows) illustrates how artifact removal with ATf, STf, and PT25, and by an expert affected average NREM sleep power density spectra of a single subject.

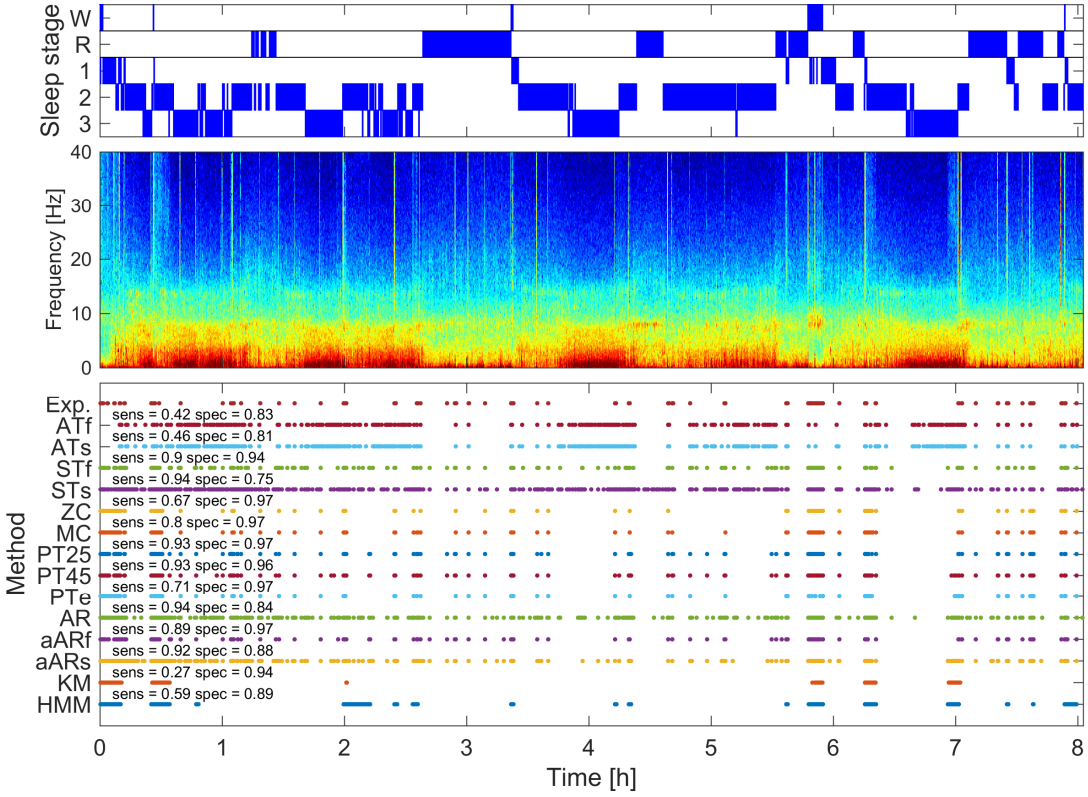


Figure 2.2. Example of artifact detection in a recording of dataset 1. Top: hypnogram (W: waking, R: REM sleep, 1 to 3: NREM sleep stages N1 to N3). Middle: spectrogram (power density spectra of 20-s epochs color-coded on a logarithmic scale [0 dB = 1 μV2/Hz; -10 dB 20 dB]). Bottom: artifacts marked by an expert (Exp.) and artifacts determined by the different algorithms (see Table 2.1 for meaning of abbreviations). Dots corresponding to 20-s epochs marked as an artifact. Note that due to the condensed display, dots may overlap. Sensitivity (sens, TPR) and specificity (spec, 1-FPR) achieved by the different algorithms are indicated

Artifact removal affected mainly frequencies above 16 Hz. The algorithms excluded generally more epochs (approximately twice as many) than an expert, as the parameters were derived with a FPR set at 10 %. In the case of artifact exclusion with ATf, STf and PT25, variability of the average spectra (standard deviation) was smaller than after artifact exclusion by

experts (Figure 2.5). Artifact exclusion mainly resulted in reduced power density in frequencies above 16 Hz. With ATf less high frequency artifacts were removed than by expert scoring (Figure 2.5 bottom, red curve above green one) while with STf and PT25 more high frequency artifacts were removed than by the expert marking. However, there was no difference in frequencies below 16 Hz.

Due to the large inter-individual differences, we were not able to find statistically significant differences in average power density spectra between no artifact exclusion and artifact exclusion by algorithms or an expert. How close average spectra match between expert marking and algorithms may be another benchmark to assess the quality of an algorithm.

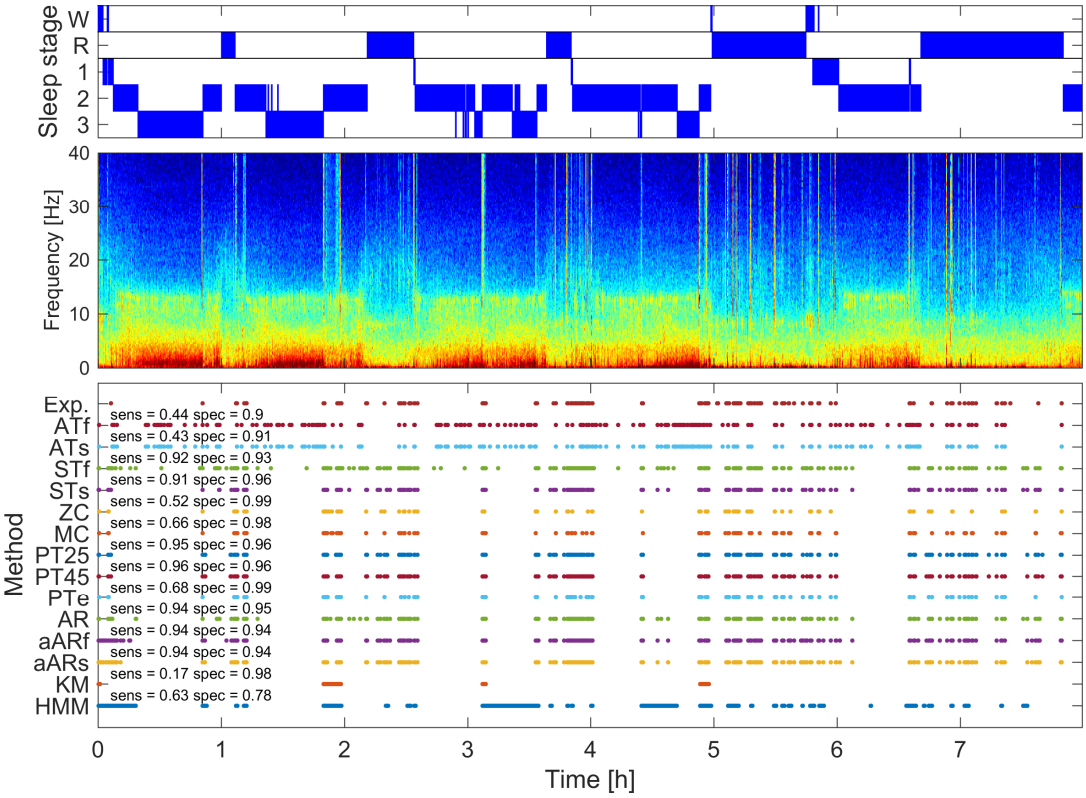


Figure 2.3. Further example of artifact detection in a recording of dataset 1. For details see Figure 2.2

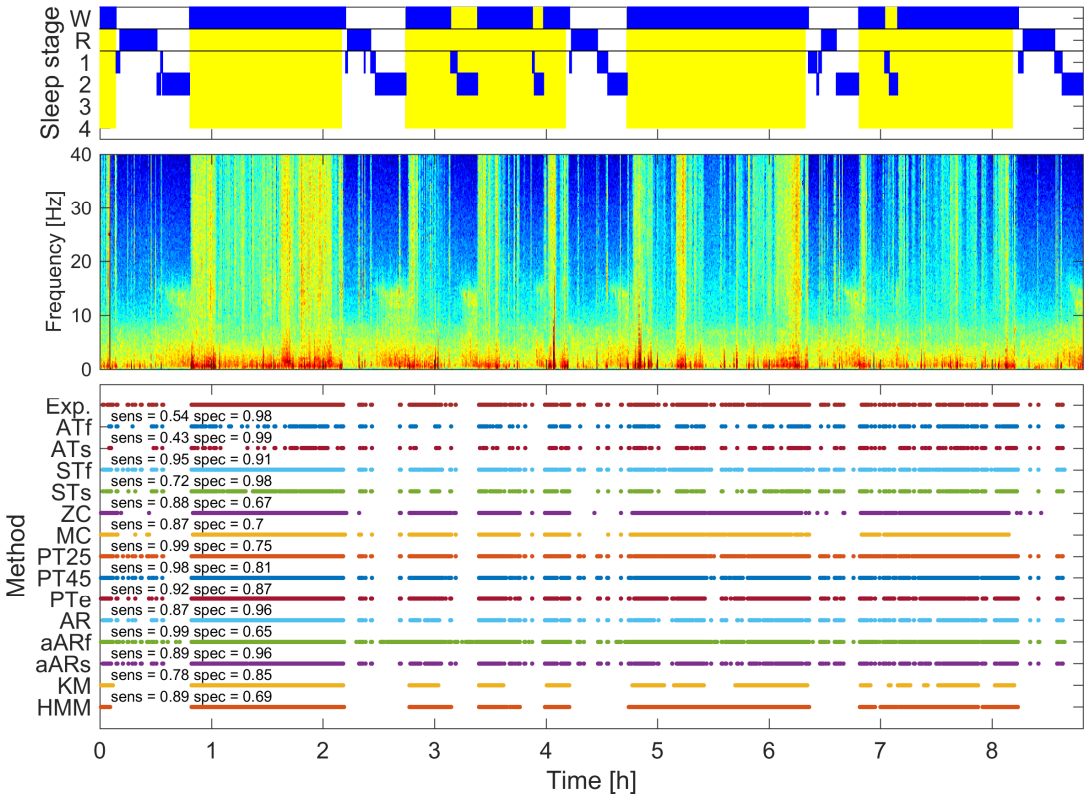


Figure 2.4. Example of artifact detection in a MSLT recording of dataset 2. Top: hypnogram (W: waking, R: REM sleep, 1 to 4: NREM sleep stages 1 to 4). Middle: spectrogram (power density spectra of 30-s epochs color-coded on a logarithmic scale [0 dB = 1 μV^2/Hz; -10 dB  20 dB]). Bottom: artifacts marked by an expert (Exp.) and artifacts determined by the different algorithms (see Table 2.1 for meaning of abbreviations). Dots corresponding to 30-s epochs marked as an artifact. Note that due to the condensed display, dots may overlap. Sensitivity (sens, TPR) and specificity (spec, 1-FPR) achieved by the different algorithms are indicated

2.5 Discussion

We performed a systematic evaluation of mostly simple algorithms that can easily be implemented and demonstrated that they work well reaching moderate to good sensitivity (TPR) while specificity (1 - FPR) was fixed at 0.9. A

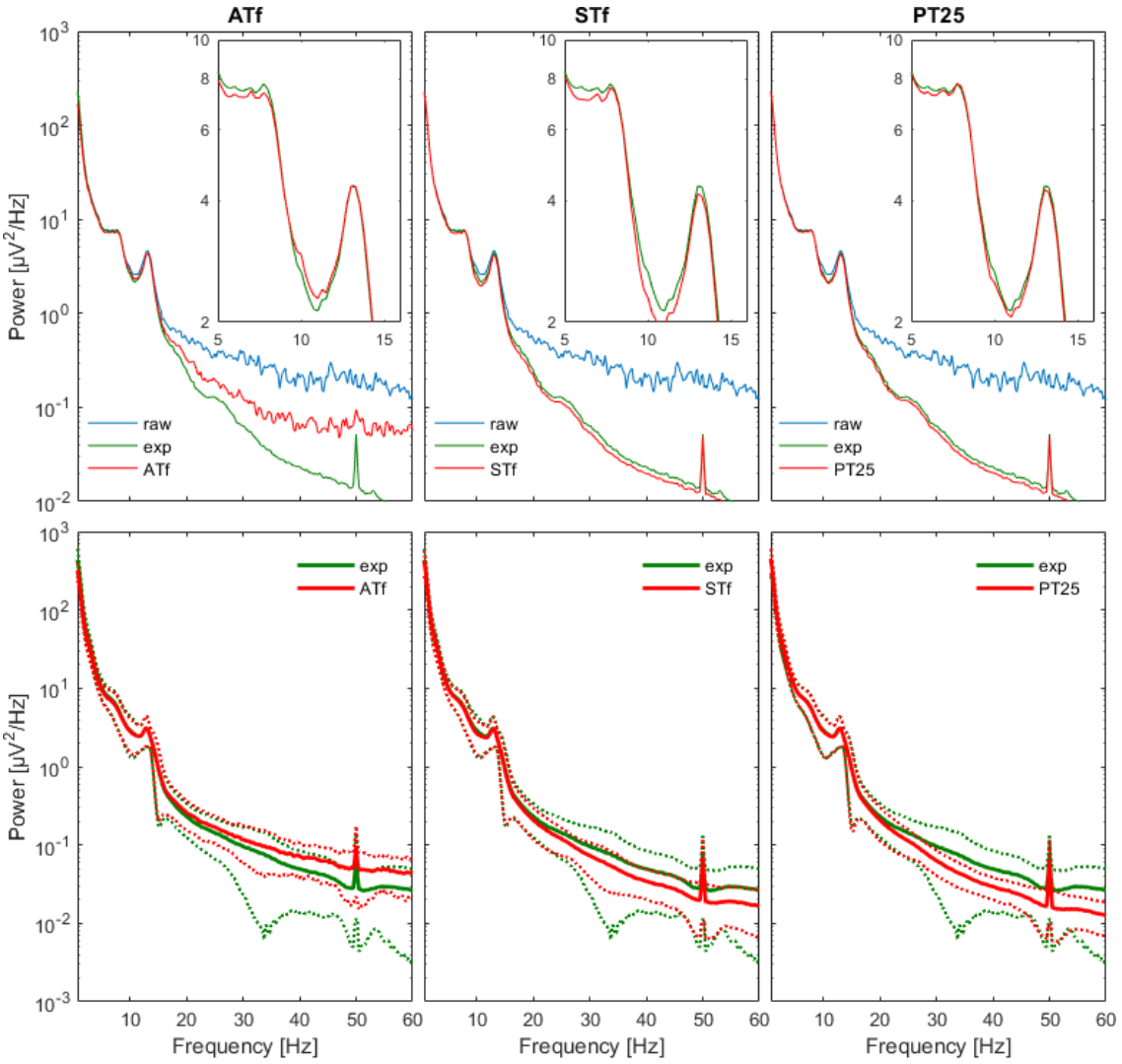


Figure 2.5. Impact of artifact exclusion on average power density spectra. Top rows: average power density spectra of NREM sleep of one night of a single subject. No artifact exclusion (blue), artifacts excluded by an expert (green) and artifacts excluded by algorithms (red). Inset illustrates spectra between 5 and 16 Hz. Bottom rows: average power density spectra of NREM sleep across subjects (n=21) of the test data set. Dashed lines show standard deviations. Only spectra after artifact removal are shown. First column: amplitude thresholding, fixed threshold (ATf). Second column: slope thresholding, fixed threshold (STf). Third column: Power thresholding 25-90 Hz (PT25). With ATf less high frequency artifacts were removed than by expert scoring (red curve above green one) while the two other methods removed more high frequency artifacts than by expert scoring

recent paper of a specific algorithm reported specificity of approximately 0.95 (D'Rozario et al., 2015) and Durka et al. (2003) observed FPRs ranging from 0.04 to 0.14 between different raters and of 0.06 and 0.08 in one rater who scored artifacts twice at an interval of 3 weeks. The evaluated methods showed good precision to obtain clean average power density spectra as an example of quantitative EEG analyses. Our aim was not to identify particular types of artifacts like e.g. contamination by eye movements but to establish a reliable procedure to exclude artifacts to be able to obtain reproducible clean quantitative EEG measures as e.g. mean power density spectra, circumventing manual artifact scoring which is time consuming and to some degree subjective (Anderer et al., 1999, Coppieters't Wallant et al., 2016). Many previous papers focused on a specific algorithm (D'Rozario et al., 2015, Coppieters't Wallant et al., 2016) or reviewed approaches more generally not assessing their performance or did not provide parameters that could be applied (Ktonas et al., 1979, Barlow, 1983, Barlow, 1984, Barlow, 1986, Bodenstein and Praetorius, 1977, Gotman et al., 1981, Durka et al., 2003).

The estimated thresholds (parameters) of the algorithms (provided in Table 2.2) were robust and did not suffer from overfitting as the results were not specific for the dataset used for parameter estimation. Overfitting is a phenomenon when an algorithm learns properties of specific subset of the data and despite the excellent performance on the training data it shows bad performance on the new data. The tradeoff between the number of parameters and quality of the fit is called bias-variance tradeoff (Geman et al., 1992) and should be taken into account. The thresholds could be applied to independent datasets and data of another laboratory with different types of recordings (sleep and MSLT) reaching the same performance as with the training dataset. However, we studied adult EEG data, and in particular

derivation C3A2. Thus, for different derivations or applications in children or infants the thresholds may have to be adapted, in particular for amplitude and slope thresholding.

As we demonstrated, even very simple methods can provide a good performance which is suited for practical applications. However, in the context of a particular application, a tradeoff between excluding too much data and not excluding enough artifacts needs to be found. For example, algorithms which capture high frequency features showed the best performance. However, for sleep applications focusing on the slow wave or spindle frequency range, these algorithms might exclude too much of the data as many artifacts mainly affect power density spectra above 20 Hz (Goncharova et al., 2003) (Figure 2.5). For such applications, we recommend decreasing the sensitivity of the algorithms. Additionally, using a combination of different features to detect artifacts may improve the performance (Coppieters't Wallant et al., 2016).

We developed additionally two non-supervised methods for artifact detection, which did not require predefined parameters. For this purpose, we employed HMM and K-means clustering to dissociate clean EEG from artifacts. However, these two algorithms worked well only with enough wake (artifacts due to movements) and sleep data as in the case of continuous MSLT recordings over 9 h. If only a small percentage of the recording is contaminated by artifacts as in the night recordings, the clustering turned out to be not reliable.

We excluded entire scoring epochs (20 or 30 s) whenever an artifact was detected. For sleep EEG recordings, this approach leaves enough data for subsequent analyses. However, this might not be the case for shorter wake EEG recordings. In general, the algorithms work equally well with shorter epochs

and can thus be easily adapted to the needs of the analyses. However, an important finding was that it is preferable to detect artifacts with a high temporal resolution: if we compute for example power in the high frequency range averaged across an epoch, power of an epoch with an artifact will not be very different from a clean one in case the artifact spans only over a short interval and thus contributes little to the calculated power. If we compute power on a finer time scale, then data points belonging to an artifact will stand out compared to clean areas. Similarly, human scorers often mark artifacts which span much less than the length of an epoch. Note that the used algorithms don't rely on the length of the scoring epoch. Most algorithms detect artifacts on a sample basis. If single outlier was detected, then the whole epoch was marked as an artifact. Some work on scoring epochs but the derived values are independent of the specific epoch length. Thus, length of an epoch is not relevant for the artifact detection. This also indicates that these algorithms can be applied on a finer time scale than 20 or 30 s.

We focused on methods which can be applied to single EEG derivations and do not need prior scoring of sleep stages and performed artifact exclusion. They should easily be applicable to large datasets (Luca et al., 2015) as needed to address question in genetics, epidemiology or precision medicine. Applying one of these algorithms will tremendously reduce analysis time compared to a standard approach (manual scoring). EEGLAB (Delorme and Makeig, 2004, Winkler et al., 2011) provides a large palette of tools to remove artifacts like eye blinks or ECG contamination, but is based on multi-channel (> 30) EEG recordings.

We used different measures to assess the performance of the algorithms, among them sensitivity and specificity, area under the ROC curve and average power density spectra. For our applications we selected as optimal

parameters those that corresponded to a FPR approximately 0.1. Even when FPR was set at 0.1, the observed values varied considerably showing the expected values on average only (Table 2.2). It should be noted however, that optimizing one performance measure does not imply that all the other ones are optimized simultaneously. Thus, one needs to decide which aspect has to be optimized.

Although we demonstrated that automatic artifact detection and exclusion works well, in the first place one should aim at obtaining high-quality EEG recordings avoiding as many artifacts as possible.

2.6 Conclusion

The study demonstrated that simple algorithms work well to automatically detect artifacts in EEG recordings in healthy participants and patients reaching good sensitivity and specificity. They are easily applicable to large datasets and will speed up data processing tremendously. Many of them even work for on-line data processing and might thus be useful in applications like “closed loop” stimulation during sleep (Ngo et al., 2013, Fattinger et al., 2017).

2.7 Acknowledgements

The study was supported by a grant of nano-tera.ch (20NA21_145929), of the Swiss National Science Foundation (32003B_146643), the ETH Zurich Research Grant ETHIIRA (ETH-18 11-1), and the NCCR Transfer Projects of the Swiss National Science Foundation (51NF40-1444639).

2.8 Supporting Information: Algorithms

The implemented algorithms (Table 1) and the corresponding parameters (derived and applied are listed in Table 2) are described below.

2.8.1 Amplitude thresholding (ATf, ATs) (Cluitmans et al., 1993)

One of the simplest approaches is to set a threshold for the absolute value of the signal amplitude and label samples, which exceed this threshold as artifacts. Thresholds may be at a fixed level for all recordings (ATf, f stands for fixed threshold), then the algorithm works online or derived from statistical properties of the signal (ATs, s stands for statistical threshold), for example in proportion to the standard deviation of the signal. The latter is not applicable online. We computed the standard deviation in each recording based on the entire recording. It is meant to overcome limitations of fixed threshold due to inter-individual variability.

2.8.2 Slope thresholding (STf, STs) (Barlow, 1983)

Another basic method we employed is a slope thresholding. We set a threshold for the absolute value of the slope (i.e. 1^{st} derivative of the EEG in μV/s). It is as in the previous case either at a fixed value for all recordings (STf) or a value based on the standard deviation of the slope of the entire recording (STs).

2.8.3 Zero crossings (ZC) (Smith et al., 1975)

We calculated the number of zero crossings in a moving window of 1 s and labeled the central point as an artifact in case the number of zero crossings in the window exceeded a certain threshold, which means that the dominating

frequency is above a certain level. For a pure harmonic signal, the half the number of zero crossings in a 1-s window is equal to the frequency of the signal.

2.8.4 Mean crossings (MC) (Smith et al., 1975)

The method is similar to the previous one, but we subtract from the signal the average of the moving 1-s window. The idea behind this approach is to not take into account slow fluctuations, as it often happens that high-frequency activity is superimposed on low-frequency activity and the amount of zero crossings is low even though high frequency activity is present. The purpose is to detect higher frequency contaminations.

2.8.5 Power thresholding (PT25, PT45, PTe) (Gotman et al., 1981, Goncharova et al., 2003)

We computed power in the high frequency range and thresholded it as movement and muscle artifacts show up at frequencies above 25 Hz (Goncharova et al., 2003).

We applied two methods for the power calculation:

1) We bandpass filtered the signal (Butterworth filter of order 10), then computed the square value of every sample. Squared amplitudes above a threshold were considered as artifacts. We applied this approach in two frequency bands 25-90 Hz (PT25) and 45-90 Hz (PT45). A notch filter at 50 Hz and 100 Hz was additionally applied. Notch filter at 100 Hz was used only in the recordings with the sampling rate higher than 200 Hz.

2) We computed power in the high frequency range (25-90 Hz) of 20- or 30-s epochs (PTe, e stands for epoch) and thresholded the average power

values per epoch. Power density spectra of 20- or 30-s epoch were computed as described in Methods.

2.8.6 Autoregressive Model (Inverse filtering; AR) (Bodenstein and Praetorius, 1977)

This method is based on the idea that EEG may be represented as a result of applying a filter to white noise (AR model). One can estimate parameters of a filter and construct an inverse filter, thus if we apply an inverse filter to EEG signal we are supposed to obtain white noise in case of a clean EEG. If an EEG has artifacts, the results of inverse filtering will deviate from white noise. Thus, one can apply a threshold to the amplitude of the resulting noise to detect artifacts (Schlögl, 2000). The entire recording is needed to derive the distribution of the residuals of the AR model.

We applied the function (detectmuscle) of the BioSig toolbox (Schlögl and Brunner, 2008) an AR model of order p=10 and a threshold proportional to the standard deviation of the noise (TH*sigma, where TH is a parameter of the function). We used the code provided by Alois Schlögel (Schlögl, 2000) with a slight modification, we changed the default threshold with the one provided by the user.

2.8.7 Adaptive autoregressive modeling (aARf, aARs) (Schlögl, 2000, Schlögl A., 1999, Schlögl et al., 1997)

The idea of this method is similar to the previous one, but it may work online (fixed threshold aARf). We use autoregressive model to predict every next EEG sample, then we compute the difference between predicted sample and the measured value; this difference is called prediction error. Artifacts were detected when the prediction error was higher than a certain threshold.

We used fixed thresholds for all recordings (aARf) or proportional to the standard deviation across a recording (aARs). In aARf, f stands for fixed and s in aARs for statistical derived thresholds. The advantage of the fixed threshold is that we do not need the entire recording to derive the threshold, thus it can work online. In this approach, model parameters were updated on every time step. The coefficients of the AR model can be updated using a Kalman filter (Schlögl, 2000). We used the function of Alois Schlögl (Schlögl, 2000) (http://pub.ist.ac.at/~schloegl/matlab/aar/aar.m; last accessed 22.06.2016) and we applied the recommended parameters (Schlögl, 2000).

2.8.8 K-means (KM) clustering

We computed power in the 40-90 Hz range of each epoch and applied a K-means clustering algorithm (Lloyd, 1982) with the number of clusters K equal to 2. Epochs were clustered into two categories: clean epochs and epochs with artifacts. This method worked well for MSLT recordings, which contained many epochs with artifacts (basically all the waking epochs where patients could move in-between measurements) and did not work for sleep recordings, which contain only a small amount of epochs with artifacts. We used MATLAB tools to perform K-means clustering.

2.8.9 Hidden Markov Model (HMM)

We also applied a Hidden Markov model (Stratonovich, 1960) to the time series of power in the 40-90 Hz range. We assumed that the power is the observable variable and there are two hidden states: an artifact and no artifact in the signal. We computed mean power in the 40-90 Hz frequency range and determined the minimum and maximum value across the recording. This range was subdivided into 64 bins and the Baum-Welch algorithm (Welch, 2003) was

used to estimate transition and emission matrices with an initial guess derived from K-means clustering. We applied the Vitterbi algorithm (Viterbi, 1967) to infer the most probable sequence of hidden states (artifact, no artifact). Results of this method were similar to the K-means algorithm, i.e. it worked well for the data with a large number of epochs contaminated with artifacts (waking) and did not work for the data with small amounts of contaminated epochs.

3 Automatic human sleep stage scoring using Deep Neural Networks

Alexander Malafeev[1,2,3], Dmitry Laptev[4], Stefan Bauer[4], Ximena Omlin[2,5], Aleksandra Wierzbicka[6], Adam Wichniak[7], Wojciech Jernajczyk[6], Robert Riener[2,3,5], Joachim Buhmann[4] and Peter Achermann[1,2,3]

[1] Chronobiology and Sleep Research, Institute of Pharmacology and Toxicology, University of Zurich, Switzerland

[2] Neuroscience Center Zurich, University of Zurich and ETH Zurich, Switzerland

[3]Center for Interdisciplinary Sleep Research, University of Zurich,Switzerland

[4] Information Science and Engineering, Institute for Machine Learning, ETH Zurich, Switzerland

[5] Sensory-Motor Systems Lab, ETH Zurich, Switzerland

[6] Sleep Disorders Center, Department of Clinical Neurophysiology, Institute of Psychiatry and Neurology in Warsaw, Warsaw, Poland

[7] Third Department of Psychiatry and Sleep Disorders Center, Institute of Psychiatry and Neurology in Warsaw, Warsaw, Poland

Disclosure: The authors declare no conflicts of interest.

Author contributions: AM, DL and PA designed the analyses; AM conducted the analyses; XO, RR, and PA; AW, AW, and WJ collected the data; DL gave extensive comments on the manuscript; SB and JB provided computational resources and consultations on the methods; AM and PA wrote the manuscript and all authors commented and accepted the final version.

3.1 Abstract

A first step in the quantitative analysis of polysomnographic data is the classification of sleep stages. Sleep stage scoring heavily relies on the visual pattern recognition by a human expert. Since sleep scoring is time consuming and partially subjective there is a need for automatic classification. In this work we developed various machine learning algorithms for sleep classification: random forest classification based on features and artificial neural networks working both with features and raw data. We tested our methods on healthy subjects and on the patients. Most methods yielded good results, comparable to interrater agreement. Our study revealed that deep neural networks performed better than feature-based methods. We also demonstrated that it is important to take the local temporal structure of sleep into account.

3.2 Introduction

3.2.1 Problem statement

Visual scoring of the sleep stages is the gold standard in sleep research and medicine. Sleep scoring is performed visually based on the following signals: (1) electrical activity of the brain - electroencephalogram (EEG), (2) electrical activity resulting from the movement of the eyes and eye lids – electrooculogram (EOG) and (3) muscle tone recorded under the chin (submental) – electromyogram (EMG).

Sleep scoring is usually performed according to one of the two standardized scoring rules: Rechtschaffen and Kales (Rechtschaffen and Kales, 1968) or American Association of Sleep Medicine (AASM) (Iber et al., 2007). According to the AASM rules (Iber et al., 2007) an expert visually classifies consecutive 30-s epochs of polysomnographic (PSG) data (EEG, EOG and EMG)

into wake, rapid eye movement (REM) sleep, and non-REM (NREM) sleep (stages N1 to N3). If scoring is performed according to Rechtschaffen and Kales (1968), 20- or 30-s epochs are scored and NREM sleep is subdivided into stages 1 to 4 with stage 3 and 4 considered as slow wave sleep (SWS, corresponding to N3 and N4). Another difference is that Rechtschaffen and Kales (1968) defined additionally movement time as a separate stage.

A plot of the sequence of sleep stages is called a hypnogram (see Figure 3.1). Human sleep starts generally with a stage 1 (N1), which usually lasts only up to several min. It is a very light sleep and one may wake up easily, even from a slight noise. Slow rolling eye movements are a feature of stage 1 and contractions of the muscles, hypnogogic jerks may occur.

Next follows stage 2 (N2). This is a state of deeper sleep than stage 1 and it is characterized by the occurrence of sleep spindles and K-complexes and an intermediate muscle tone.

Stage 2 usually precedes deep sleep – stage 3 and 4 (SWS, N3). The main characteristic of deep sleep is the presence of slow oscillations (< 1 Hz) and delta waves (1-4 Hz) in the EEG for more than 20 % of an epoch. The muscle tone is low.

REM sleep occurs periodically throughout the night and is characterized by rapid eye movements, fast low-amplitude EEG activity similar to the wake EEG, and a low muscle tone (atonia).

The progression of the different stages is not random, but rather follows a cyclic alternation of NREM and REM sleep (Achermann and Tarokh, 2014) with a cycle duration of approximately 90 min (see Figures 3.1 for a typical structure). Healthy sleep consists of approximately 3-5 sleep cycles.

Visual scoring by an expert is time consuming and to some degree subjective. Several studies addressed the interrater reliability and revealed

that correspondence between scorers is far from ideal (Danker-Hopfe et al., 2004, Penzel et al., 2013, Rosenberg and Van Hout, 2013, Younes et al., 2016, Younes et al., 2018). Cohen kappa values in Danker-Hopfe study showed strong agreement for REM sleep, minimal agreement for stage 1 and moderate agreement (McHugh, 2012) for the other stages.

Shortly after the sleep scoring standard was established in 1968 (Rechtschaffen and Kales, 1968), attempts were made to develop algorithms for automated sleep staging (Itil et al., 1969, Gevins and Rémond, 1987, Larsen and Walter, 1970, Smith and Karacan, 1971, Gaillard and Tissot, 1973, Martin et al., 1972).

3.2.2 Related work

Martin et al. (Martin et al., 1972) applied a simple decision tree using EEG and EOG data. A decision tree like algorithm was also used by Louis et al. (2004). Stanus et al. (1987) developed and compared two methods for automatic sleep scoring: one based on an autoregressive model and another one based on spectral bands and Bayesian decision theory. Both methods used one EEG, two EOG and an EMG channel. The EOG was needed to detect eye movements and the EMG to assess the muscle tone. Fell et al. (1996) examined automatic sleep scoring using additional non-linear features (correlation dimension, Kolmogorov entropy, Lyapunov exponent) and concluded that such measures carry additional information not captured with spectral features. Park et al. (2000) built a hybrid rule- and case- based system and reported very high agreement with human scorers. They also claimed that such a system works well to score patients with sleep disorders.

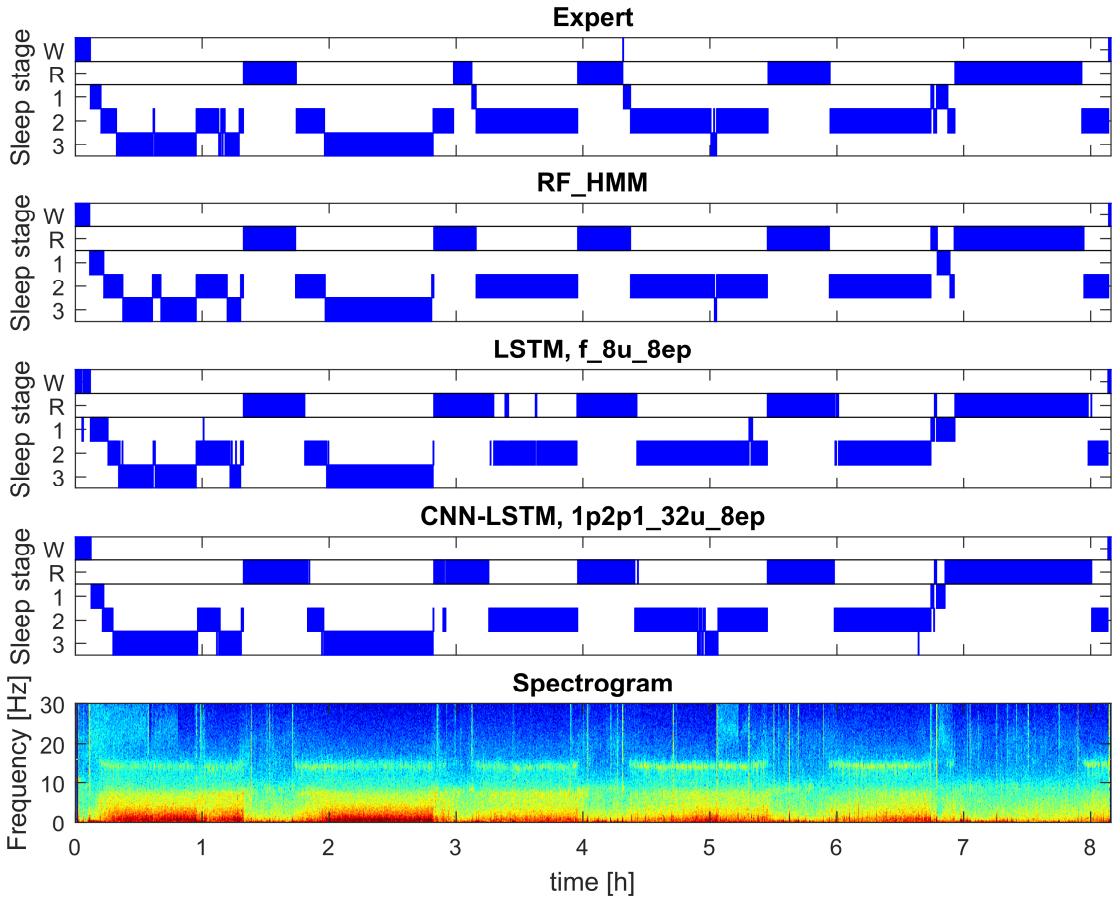


Figure 3.1. Example of automatic sleep scoring trained on healthy subjects (dataset 1; example from validation set). Panel 1: hypnogram (W: waking, R: REM sleep, 1 to 3: NREM sleep stages N1 to N3) scored by a human expert. Panel 2: hypnogram resulting from RF classification based on features followed by temporal smoothing with HMM. Panel 3: hypnogram resulting from classification with 3-layer bidirectional LSTM network with 8 LSTM neurons in each layer based on features, sequence length 8 epochs (i.e. 160 s). Panel 4: hypnogram resulting from a CNN-LSTM network with 11 convolutional layers and 2-layer bidirectional LSTM with 32 LSTM neurons in each layer. Input comprised of raw data (1 EEG and 2 EOG) and EMG power (1 value per epoch). Bottom panel: spectrogram (power density spectra of 20-s epochs color-coded on a logarithmic scale [0 dB = 1 $\mu V^2/Hz$; -10 dB  20 dB]) of EEG derivation C3A2. See supplementary material for the naming conventions of the algorithms

One of the commercially successful attempts to perform automatic scoring evolved from the SIESTA project (Klosh et al., 2001). The corresponding software of the SIESTA group was named Somnolyzer 24x7. It includes a quality check of the data based on histograms. The software extracts features based on a single EEG channel, two EOG channels and one EMG channel and predicts sleep stages using a decision tree (Anderer et al., 2005). Software was validated on a large database containing 90 patients with various sleep disorders and ~200 controls. Several experts scored sleep in the database and Somnolyzer 24x7 showed very good agreement with consent scoring (Anderer et al., 2005).

Newer and more sophisticated approaches were based on artificial neural networks (ANNs). Schaltenbrand et al. (1993) for example applied ANNs for sleep stage classification using 17 features extracted from PSG signals and reported an accuracy close to 90 %. Pardey et al. (Pardey et al., 1996) combined ANNs with fuzzy logic and Längkvist et al. (2012) applied restricted Bolzmann machines to solve the sleep classification problem, to mention just a few approaches.

The methods mentioned above require carefully engineered features. It is possible to avoid this step using novel deep learning methods. ANNs in the form of Convolutional Neural Networks (CNNs) were recently applied to the raw sleep EEG by Tsinalis et al. (2016). CNNs are especially promising because they can learn complex patterns and 'look' at the data in a similar way as a 'real brain' (Fukushima and Miyake, 1982). However, working with raw data requires a huge amount of training data and computational resources.

Several previous epochs are taken into account by a human expert according to the scoring manuals.Therefore, we assume that learning local temporal structures are an important aspect in automatic sleep scoring.

Temporal patterns have previously been addressed by applying a hidden Markov model (HMM) (Doroshenkov et al., 2007, Pan et al., 2012).

In the last few years, Recurrent Neural Networks (RNN) have demonstrated superiority over "classical" Machine Learning (ML) methods on datasets with a temporal structure (Mikolov et al., 2010, Karpathy and Fei-Fei, 2015, Graves et al., 2013). One of the most common and well-studied RNN is the Long-Short Term Memory (LSTM) neural network (Hochreiter and Schmidhuber, 1997).

Such networks have already been successfully applied to EEG data in general (Davidson et al., 2006) as well as to sleep data (Supratak et al., 2017). The results of the methods using raw data are comparable to the best outcomes of algorithms which used engineered features and classical machine learning methods (Davidson et al., 2006, Supratak et al., 2017).

The above-mentioned approaches were based on supervised learning. There have also been several attempts to perform unsupervised automatic sleep scoring in humans (Agarwal and Gotman, 2001, Grube et al., 2002, Gath and Geva, 1989) and in animals (Sunagawa et al., 2013, Libourel et al., 2015).

3.2.3 Our contribution

We implemented different machine learning algorithms (random forests, feature based networks and raw data based networks) and trained and tested them on engineered features as well as on raw data of healthy participants and patients.

We found that all our algorithms performed well on the data of healthy subjects. Performance on the data recorded in patients of another laboratory was lower, but it deteriorated less for ANNs. We found that including part of the patient data into the training improved performance on the patient data. It

suggests that we would need even larger and diverse datasets in order to train an algorithm which can be applied reliably in practice. We found that a deep neural network produced good results even using a single EEG channel. It was one of the most fascinating observations of our work.

Despite the fact that automatic scoring algorithms have shown reasonably high performance there is no consensus yet in the community that they perform well enough to replace human scorers. This paper provides comparisons of different automatic scoring algorithms validated on two different datasets, including not only healthy subjects, but also patients.

3.3 Methods

3.3.1 Polysomnographic (PSG) data

We trained and tested automatic sleep stage scoring algorithms on two datasets from two different laboratories.

The first dataset was comprised of 54 whole night sleep recordings of healthy participants. The second dataset consisted of 22 whole night sleep recordings and 21 recordings of a multiple sleep latency test (MSLT) in patients. The MSLT is routinely used to evaluate daytime sleepiness of patients. During this test a subject has four or five 20-min nap opportunities, which are separated by 1.5-hour long intervals. An example of an MSLT hypnogram can be seen in Figure 3.2. Usually, only naps are recorded, but in our dataset, recordings were continuous over approximately 9 h and occasionally we observed sleep episodes in addition to the scheduled naps. In a standard setting these sleep episodes would have been missed.

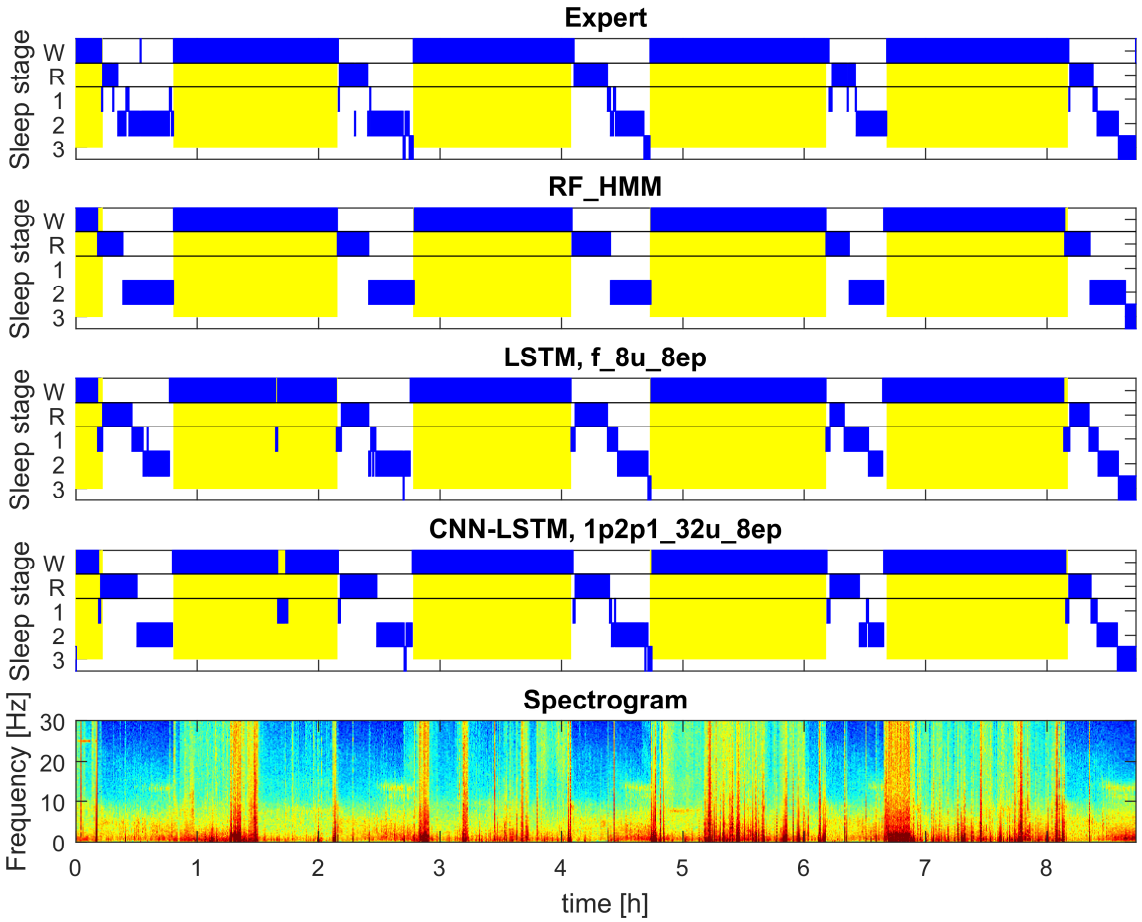


Figure 3.2. Example of automatic sleep scoring of MSLT data trained on a mixture of data of healthy participants and patients data (dataset 1 and 2; example of test set). Figure structure and abbreviations are analogous to Figure 3.1. Yellow background represents lights on

Dataset 1: Healthy subjects

Polysomnographic (PSG) recordings from a study investigating the effect of vestibular stimulation (Omlin et al., 2018). In total 18 healthy young males (20-28 years; mean: 23.7 years) were recorded. Three nights of sleep (8 h) were recorded in each subject. Two nights with motion (bed was rocked till sleep onset or for the first 2 h after lights off), and a control night without movement. Data were composed of 12 EEG channels, applied according to 10-20 system, 2 EOG derivations, 1 submental EMG derivation, 1 ECG derivation and respiration signals (chest and abdomen). Recordings were performed with

a polygraphic amplifier (Artisan, Micromed, Mogliano, Veneto, Italy). Sampling rate was equal to 256 Hz (Rembrandt DataLab; Version 8.0; Embla Systems, Broom Field, CO, USA). A high pass filter (EEG: -3 dB at 0.16 Hz; EMG: 10 Hz; ECG: 1 Hz) and an anti-aliasing filter (-3 dB at 67.4 Hz) were applied to the analogue signals. The EEG derivations were re-referenced to the contra-lateral mastoids (A1, A2). Sleep stages (20-s epochs) were scored according to the AASM criteria (Iber et al., 2007). The study was performed in the sleep laboratory of the Institute of Pharmacology and Toxicology at the University of Zurich, and was approved by the Institutional Review Board of the Swiss Federal Institute of Technology in Zurich (ETH Zurich).

Dataset 2: Patients

Data were recorded in patients with narcolepsy (23 patients) and hypersomnia (5 patients) during a night of sleep (approx. 8 h) and during a multiple sleep latency test (MSLT, continuous recordings over approx. 9 h). We had to exclude some recordings due to bad signal quality. Thus, some patients contributed only with a night or a MSLT recording (Hypersomnia: 5 MSLT, 4 nights; Narcolepsy: 16 MSLT, 18 nights). Data were comprised of 6 EEG, 2 EMG, 2 EOG derivations and 1 ECG. Signals were recorded at a sampling rate of 200 Hz (polygraphic amplifier Grass Technologies AURA PSG). A high pass filter (EEG: -3 dB at 0.5 Hz) and an anti-aliasing filter (-3 dB at 50 Hz) were applied to the analogue signals. Sleep stages (30-s epochs) were scored according to Rechtschaffen and Kales (1968). Movement time was not scored. To make sleep stages compatible with the first dataset, we merged sleep stages 3 and 4. Recordings were performed at the Sleep Disorders Center, Department of Clinical Neurophysiology, Institute of Psychiatry and Neurology in Warsaw,

Warsaw, Poland. The study was approved by the Institutional Review Board of Institute of Psychiatry and Neurology.

3.3.2 Machine Learning: classification

Machine Learning is a branch of computer science which allows to learn properties of the data and solve problems without direct programming of the decision rules. The main problems which can be solved with machine learning are regression, clustering and classification (Bishop, 2016). Classification algorithms solve the problem of assigning labels to the data. The algorithms are trained with labeled data, the training set, to learn properties of the data and the corresponding labels (supervised machine learning (Bishop, 2016).

In this work, we solved the classification problem by applying supervised machine learning algorithms. We followed two approaches, 1) classification based on features (random forest (RF) and artificial neural networks (ANNs)) and 2) classification based on raw data (ANNs).

Classification based on features

Polysomnographic signals are very complex, but they reveal certain patterns crucial for scoring by an expert. For example, waves of certain frequencies: sleep spindles (12-14 Hz), slow waves (0.5-4 Hz), alpha waves (10 Hz), theta oscillations (4-8 Hz) are very important to distinguish the different sleep stages. These measures can be easily quantified in the frequency domain. Other important markers of sleep stages such as rapid and slow eye movements, eye blinks and muscle tone can also be quantified. Such measures are called features and the process of their definition is called feature engineering. Using carefully engineered domain-specific features for machine learning systems has a lot of advantages: it requires a small amount of training

data, is fast and the results are interpretable. Another approach based on deep learning, working with raw data, is described later.

Preprocessing and feature extraction

In a first step, we used spectrograms of the EEG instead of using the raw signal. It is well known that spectra capture the major properties of the sleep EEG and this way we were able to significantly reduce the dimensionality of our data. Power density spectra were calculated for 20-s epochs (30-s for patient data) using the Welch function in MATLAB (FFT; average of four or six 5-s windows; Hanning windows; no overlap; frequency resolution 0.2 Hz). Spectra were plotted and color-coded on a logarithmic scale (Figures 3.1 and 3.2). Spectrograms were limited to the range of 0.8-40 Hz to reduce the dimensionality of the data matrix.

We used a set of 20 engineered features for the classification (see Supplemental material for their definitions). They include among others power in different frequency bands and their ratios, eye movements, and muscle tone. We did not exclude any epochs (i.e. included artifacts), because we wanted to have a system, which is ready to work with the data with a minimal requirement of manual preprocessing. Moreover, epochs with artifacts contain useful information: wakefulness is almost always accompanied by movement artifacts and a movement is often followed by a transition into stage 1.

We used two different approaches for the classification based on features: random forest (RF) and artificial neural networks (ANNs).

Random Forest (RF)

One of the classical methods to solve classification problems is based on decision trees (Morgan and Sonquist, 1963, Hunt et al., 1966, Breiman et al.,

1984). Every node of a tree corresponds to a feature and a corresponding a threshold value. For a data vector which has to be classified, we traverse the tree by comparing a corresponding feature to the threshold of the node. Depending on the outcome of the comparison, we go to the left or to the right branch. Once we have traversed the tree, we end up in a leaf that determines to which class the data point belongs to.

Decision trees have certain limitations (e.g. overfitting) (Safavian and Landgrebe, 1991, Mitchell, 1997). Overfitting means that an algorithm learns something very specific of the training data and the classifier can no longer predict new data.

A way to overcome these limitations is to create an ensemble of trees: i.e. to build many trees**,** each based on a random subset of the training data (Ho, 1995, Breiman, 2001). A data point is classified by all trees and we can compute the probability of a data point belonging to a particular class by the fraction of trees which "voted" for this class. RF classifiers and similar recent tree-based technique demonstrated state-of-the-art results on a variety of problems (Chen and Guestrin, 2016, Laptev and Buhmann, 2014, Laptev and Buhmann, 2015).

We implemented the RF to classify sleep stages based on feature vectors (20 components). We computed probability vectors for every epoch (20 or 30 s). Further we considered the temporal structure of sleep as described above about time course learning. We applied a hidden Markov model (HMM; see supplementary material) and a median filter (MF) with a window of three 20-s or 30-s epochs to smooth the data.

Artificial neural networks (ANNs)

For a long time, researchers have been trying to build a computer model of a neuron (Farley and Clark, 1954, Rochester et al., 1956) and use such models for data classification (Rosenblatt, 1958). This research resulted in the development of multilayer neural networks (Ivakhnenko and Lapa, 1967) which are now denoted artificial neural networks (ANNs).

ANNs consist of interconnected neurons. Every neuron performs multiplication of input signals with parameters called weights, summed up and sent to the output. One can train ANNs by adjusting (updating) the weights (Goodfellow et al., 2016). This process of training is also called optimization. ANN training requires a function which quantifies the quality of the classification. Such a function is called the loss function or cost function. The loss function must be differentiable, otherwise it is not possible to compute the gradients. An example of a loss function is the mean square error. In our work, we used the cross-entropy loss function (De Boer et al., 2005). Cross-entropy loss is a good measure of errors of networks with discrete targets. Targets are the ground truth values given by an expert, in our case sleep stages.

3.3.3 Deep learning with raw data

Deep neural networks (DNNs) can learn more complex models. Moreover, DNNs can automatically learn features and the feature engineering step can be omitted. Features can be learned using, for example convolutional neural networks (Fukushima and Miyake, 1982, Waibel et al., 1989, LeCun et al., 1989). Deep neural networks usually show better performance than feature-based methods, but it comes at the price of an increased computational demand and such networks require more training data.

Convolutional Neural Networks (CNNs)

CNNs were initially developed for image recognition (Fukushima and Miyake, 1982, Waibel et al., 1989, LeCun et al., 1989). The main property of CNNs is that they perform a convolution of an input with a set of filters, which have to be learned. They were successfully applied not only for image recognition, but also in speech recognition (Abdel-Hamid et al., 2014), text analysis (dos Santos and Gatti, 2014) and many other areas. Moreover, CNNs have already been successfully applied to various types of physiological signals, including wake EEG recordings (Cecotti and Graeser, 2008l, Mirowski et al., 2008). The filters have a certain size. Given the one-dimensional nature of our data, a filter is a vector of a specific length. The filter slides with certain step called a stride across the input data.

Another specific type of layers we used was max-pooling. It takes the maximal value of the sliding window and helps to achieve local invariance. The max-pooling layer also has a specific filter size and a stride.

Residual Networks

Residual Networks (He et al., 2016) are a special kind of ANNs where layers are connected not only in sequential order but also with so-called skip or residual connections which jump over one or multiple layers. Gradients can vanish when networks have a lot of layers. Residual connections prevent this problem and make the training of networks more efficient and make it possible to train very deep networks with large numbers of layers.

3.3.4 Learning time dependencies

Common machine learning algorithms consider every data sample independent from the previous ones. This is the case for RF classification and

common ANNs. However, experts take information about previous epochs into account when they perform sleep scoring. Thus, it would be useful to consider some temporal information (structure) in the sleep classification algorithm.
As was mentioned in the introduction, sleep has not only a local but also a global structure, such as sleep cycles (Achermann and Tarokh, 2014). However, this global structure should not be taken into account while scoring (visual or automatic), as it might be different in pathology or during naps. Therefore, we limited the temporal memory of our models (see below), but the information of several previous epochs is still important to consider for sleep scoring. We assume that if we learn long sequences, it would bias the algorithm and such models would perform poorly on recordings where such patterns are not present, e.g. in the MSLT recordings or disturbed sleep.

We implemented the learning of temporal structures of sleep in two ways. First, we applied a Hidden Markov Model (HMM) (Stratonovich, 1960) to smooth the output of the RF classification (see supplemental material for details) and by a median filter (MF) with a window size of 3 epochs, a very simple yet efficient approach to smooth the data.

As a second approach we implemented recurrent neural networks (RNNs). RNNs receive their own output of the previous step as additional input in combination with the new data vector. Thus, RNNs take into account the temporal structure of the data. One of the most successful RNNs is the long-short term memory (LSTM) network (Hochreiter and Schmidhuber, 1997). RNNs can also use information about future epochs; in such a case they are called bidirectional RNNs.

As mentioned above, the length of the input sequences should be limited to reasonably short time intervals. We limited our algorithms to learn patterns not longer than 8 (2.8 or 4 min), 32 (10.7 or 16 min) and 128 epochs

(42.6 or 64 min). We dynamically formed batches of sequences: the beginning of each sequence was chosen randomly (i.e. sequences may intersect). This way more sequences may be used for training than by just taking them sequentially. For details about batches and their processing see supplementary material.

3.4 Study setup

3.4.1 Network architectures

We considered two types of networks:

1) Networks which used features as input (LSTM networks).
2) Networks which worked with raw data and used convolutional layers before the LSTM networks (CNN-LSTM networks).

3.4.2 LSTM networks

We implemented a network with 3 hidden layers (Figure 3.3). Each layer consisted of 8, 16, 32 or 128 LSTM units, and we also applied one- and bi-directional layers resulting in a total of 6 network configurations.

3.4.3 CNN-LSTM networks

We realized networks with 11 convolutional layers followed by two LSTM layers with 32 units (Figure 3.4).

We also used residual convolutional networks (19 layers) as outlined before, worked with different input signals (EEG, EOG and EMG) and created separate CNN networks (CNN blocks in Figure 3.4) for every input (EEG, 2 EOG). The outputs of all blocks were concatenated and fed into the LSTM

layers. There were two bidirectional LSTM layers. Each layer contained 32 LSTM units. There were batch normalization layers (Ioffe and Szegedy, 2015) before, between and after LSTM layers. Batch normalization layer rescales the input to make sure that all the values belong to the same range. We used separate CNN blocks for the two EOG channels because correlations between the EOG signals are important to distinguish the different types of eye movements. In case the EMG was included, only a single value (EMG power in the 15-30 Hz range) per 20- or 30-s epoch was considered. Thus, three input configurations were implemented: EEG only, EEG and EOGs, and EEG, EOGs and EMG (Figure 3.4) resulting in a total of 7 network configurations.

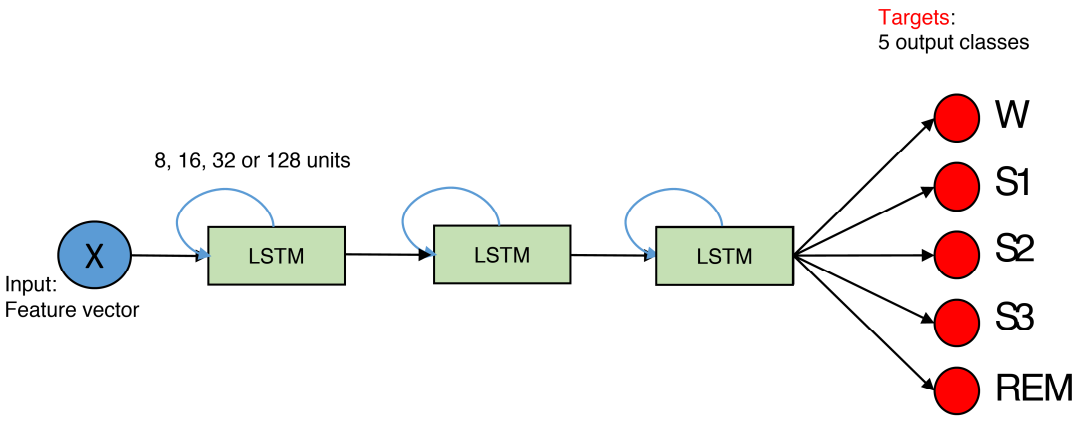


Figure 3.3. The structure of the network for feature based classification. It is composed out of 3 layers. The size of the layer is 8, 16, 32 or 128 units. Blue arrows indicate that LSTMs are recurrent. X is the input data matrix – the matrix which contains features in columns and rows correspond to epochs. In case of the spectrogram as input, it corresponds to a transposed spectrogram. Red circles depict output neurons. Their output is compared to the expert labels (targets). Every neuron corresponds to certain sleep stage (W: Wake; S1, S2, S3: NREM sleep stages; REM: REM sleep)

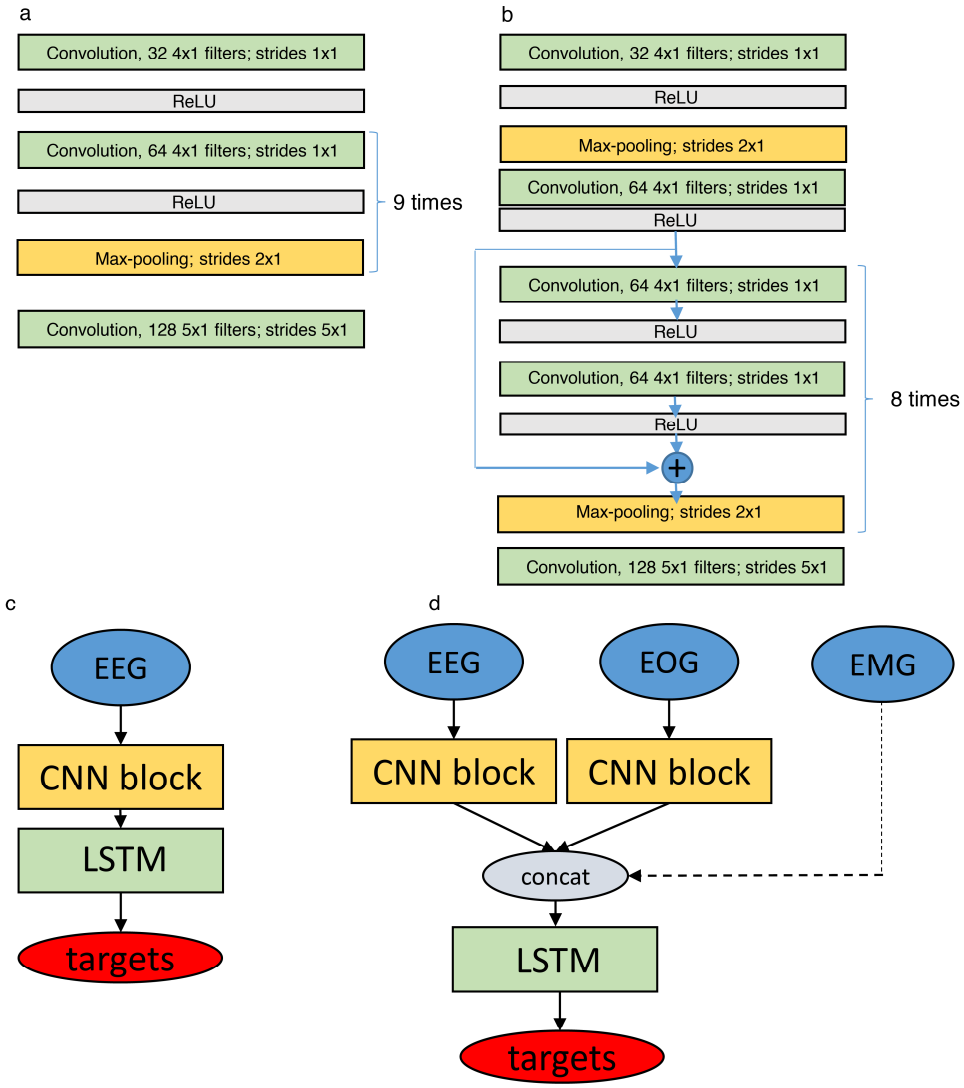


Figure 3.4. Structure of the networks for classification based on raw data. Networks have CNN and LSTM parts. **a**: CNN block (11 layers) which is used to process raw EEG and EOG data. **b:** Similar CNN block with residual connections (19 layers) **c** and **d** depict the final network structures based on the modules depicted in **a** and **b**, using only EEG data (**c**) or EEG and EOG data as input (**d**, EMG input as dashed line as it did not have a CNN block). The EMG input was a preprocessed single value (power) per epoch. LSTM networks consisted of 2 bidirectional layers with 32 units each. There were batch normalization layers before, between and after LSTM layers. Batch normalization rescales inputs to make sure they all are in the similar range. Targets are the classified sleep stages. ReLU: Rectified Linear Unit, it is an activation function to transform the activation of a neuron

3.4.4 Optimization

Networks require training which is achieved by optimization. Optimization procedures have to find minima (in case of ANN local minima) of a loss function over the parameter space (weights of the network). Weights are commonly adjusted according gradients (see supplemental material for details about optimization and regularization).

Networks were implemented using the Keras package (Chollet, 2015) with Theano (Al-Rfou et al., 2016) and Tensorflow (Abadi et al., 2016) backends. The Theano backend was used to train our feature-based LSTM networks and the Tensorflow backend to train the raw data based CNN-LSTM networks. We worked with different backends because we first developed the feature based networks and running on a desktop computer and later with raw data based networks. These networks had to be trained on GPUs and for this only the Tensorflow backend was available.

3.4.5 Training, validation, and testing

To avoid overfitting, we split dataset 1 (healthy participants) into three parts: training (36 recordings), validation (9 recordings) and testing (9 recordings). The idea was to train all our models using the training part of the data, then classify the data of the validation part and select the best models for further evaluation of their performance on the test part. Validation revealed that performance of the different models was very similar, thus it was not meaningful to select the best ones for testing. Therefore, we estimated the final performance of the algorithms with the test set. In addition, we used the whole second dataset (patients) as a test set, thus, assessing transferability of the approaches to datasets from another laboratory and to a different subject population (patients).

Further, we wanted to study how performance of the algorithms would benefit from the inclusion of patient data into the training set. We took the same training set of healthy subjects (36 recordings) and added patient data (19 recordings) to it. The remaining patient data (10 MSLT recordings and 14 sleep recordings) were used for performance evaluation together test set of the healthy participants (9 recordings). For further details see supplementary material.

3.4.6 Performance evaluation

To assess performance of our algorithms, we used the F1-score (Sørensen, 1948, Dice, 1945) as a measure of classification quality, also known as Sørensen–Dice coefficient (Sørensen, 1948, Dice, 1945), a widely used measure of classification quality with multiple classes in machine learning. The F1-score is a number in the interval from 0 to 1, with one reflecting ideal and zero bad classification.

We also computed the cross-entropy loss and accuracy (De Boer et al., 2005) during training to assess convergence of the algorithms.

3.5 Results

3.5.1 Convergence of the ANNs

To see if the networks converged, we computed cross-entropy loss and accuracy (proportion of correctly classified examples) on the training and validation datasets on every training iteration (50 iterations in total). These types of curves are called learning curves (Pedregosa et al., 2011). The learning curves corresponding to the feature-based LSTM networks are illustrated in Suppl. Figures S3.3 and S3.4 (see supplementary material for the naming

convention of the networks). Convergence is reached if cross-entropy is declining reaching a stable level and accuracy is saturating with increasing iterations.

All our networks showed good convergence when they were trained both on the data of healthy participants (Suppl. Figure S3.3) and on a mixture of both datasets (Suppl. Figure S3.4).

Learning curves for the ANN based on the raw data as input are depicted in Suppl. Figures S3.5 and S3.6. Most of the networks showed good convergence (loss monotonously decreased, and accuracy increased to saturation). Some networks showed large fluctuations of loss and accuracy on the validation set: the network which has only the EEG channel as input (1p_32u_8ep), the network which had EEG and EOG as input and 8 epoch long sequences (1p2_32u_8ep), and the network with input comprised of EEG, EOG and EMG and 128 epoch long sequences (1p2p1_32u_128ep). The least smooth learning curves were observed with the network with residual connections. This network had the largest number of parameters and thus, more data and iterations might be needed to reach convergence. We expect that such networks to perform better if trained on an extended dataset.

3.5.2 Classification performance

Figure 3.1 illustrates the hypnograms obtained with 3 selected algorithms (RF, LSTM, CNN-LSTM) in comparison with the expert scoring. In general, performance of all algorithms was good capturing the cyclic structure of sleep. Slight differences to the human scorer were observed, e.g. longer REM sleep episodes with the 3-layer bidirectional LSTM network (Figure 3.1, panel 3).

Scoring of healthy participants

The F1 scores computed on the validation part of the dataset 1 (healthy participants) are shown in the Figure 3.5 (only 4 selected methods; see Suppl. Tables 3.1 and 3.2 for F1 scores of all algorithms, validation and test data): RF classification smoothed using HMM, one LSTM network trained on features, and two CNN-LSTM networks with raw data input, one of them included residual connections.

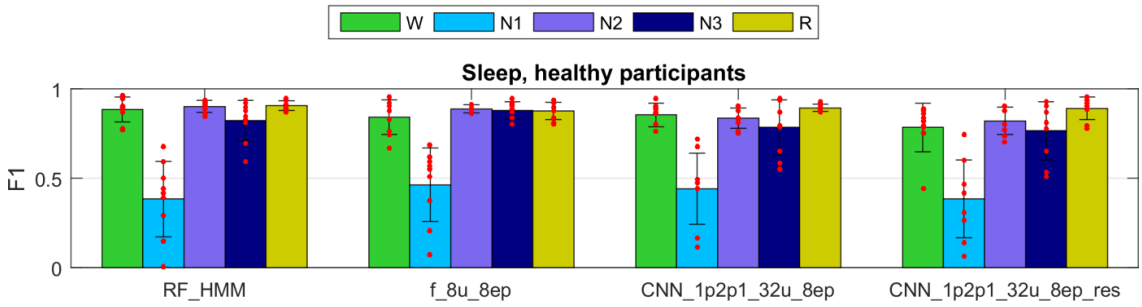


Figure 3.5. F1 scores of selected methods applied to the validation set of dataset 1 (healthy participants). The first 2 groups of bars represent feature-based classifiers (RF and LSTM network) and the last 2 groups CNN-LSTM networks based on the raw data input. See supplementary material for the naming conventions of the algorithms. Red dots represent individual F1 scores. W: wakefulness; N1 to N3: NREM sleep stages; R: REM sleep

All four methods showed high performance for all stages except for the stage 1 (N1). The F1 score for the stage 1 was around 0.4 which we still consider a good result because it is comparable to the low interscorer agreement of stage 1 (Danker-Hopfe et al., 2009, Penzel et al., 2013, Danker-Hopfe et al., 2004, Rosenberg and Van Hout, 2013).

The F1 scores of all methods evaluated on the validation part of dataset 1 are depicted in Suppl. Figures S3.7 (features) and S3.8 (raw data). Most networks performed similarly well on the validation set; those which included only a single EEG derivation as an input (Suppl. Figure S3.7, s_8u_8ep,

spectrogram as input and Suppl. Figure S3.8, 1p_32u_8ep, raw EEG as input) showed slightly lower performance, probably to the fact that the EEG spectrogram or the raw EEG do not contain information about eye movements and muscle tone. However, this was the case in some recordings only, for other recordings the performance was very good. Interestingly, performance of these networks on the test set was much better (Tables 3.1 and 3.2). We assume that validation set contained some recordings which were difficult to score using only a single EEG channel.

The network with input comprised of EEG, EOG and EMG and 128 epoch long sequences (1p2p1_32u_128ep) had a low performance on both, the validation and the test set because of large random fluctuation of accuracy in the last training iteration. Ideally, we should have stopped training of this network earlier or trained it longer.

Networks with 16 and 32 units in a layer were inferior for the scoring of stage 1 than the network with only 8 units probably due to overfitting, although the difference was very small. These networks may show a better performance if trained with larger datasets. One-directional network predicted REM sleep a bit worse than bidirectional ones. The advantage of one-directional network is the possibility to work online. Surprisingly, classification with RF smoothed with simple median filter or HMM worked almost as good as classification with ANNs (features and raw data).

Generalization to the patient data

We validated our methods on dataset 2 (patients). The F1 scores for selected methods are presented in Figure 3.6 (only 4 selected methods; see Suppl. Figures S3.9 and S3.10; Suppl. Tables 3.3 and 3.4 for F1 scores of all

algorithms used to classify patient data). Note that the data of the patient dataset were not used for the training at all.

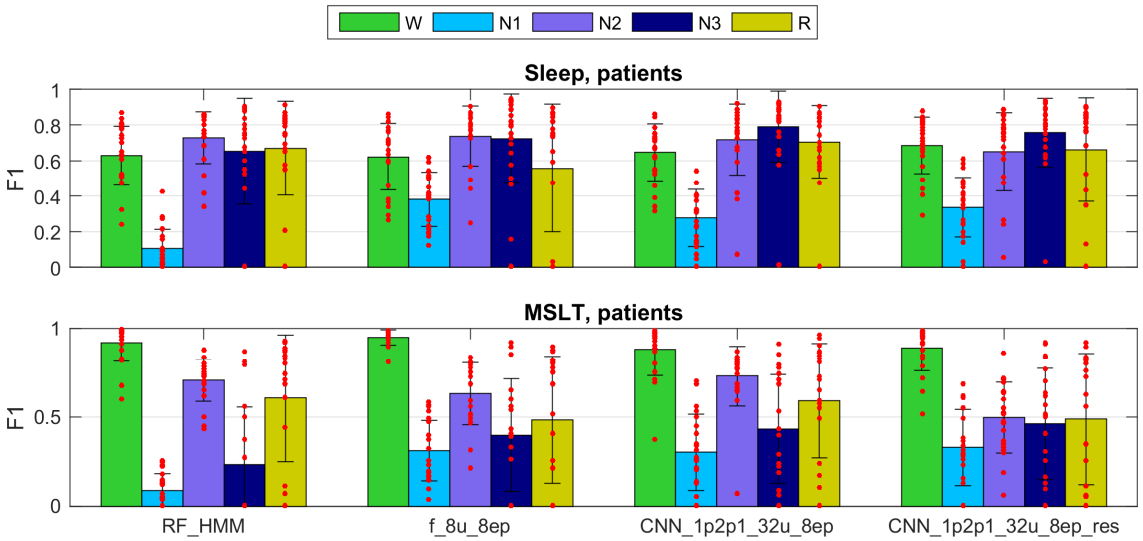


Figure 3.6. F1 scores for the same methods as in Figure 4 applied to the patient dataset. Note that the training did not include patient data. Top panel represents sleep recordings and the lower one MSLT recordings. Note that during MSLT recordings stage N3 is not always reached; such recordings were not taken into account when computing average F1 scores and standard deviations of N3. For details see Figure 3.5

The performance was somewhat lower for all classifiers applied to the sleep data of patients than in healthy participants and again lower for the MSLT data and F1 scores showed a large variance. Classification performance of stage 1 was worst for the RF classification in this dataset. F1 scores for sleep stages in the MSLT data were low since MSLT recordings contained a low proportion of sleep. The opposite holds for wakefulness, which was the dominant stage and thus, the F1 scores of wake were very high. Methods using only a single EEG signal as input (spectrogram or raw EEG channel as input) performed worse on the patient data.

Networks trained on data of both datasets

Next, we trained two networks and RF classification with a mixed training data consisting of healthy subjects (36 recordings) and part of the patient data (19 recordings; both sleep and MSLT data). We validated the models on the test part of the mixed dataset (healthy participant: 9 recordings; patients, 14 sleep and 10 MSLT recordings).

Figure 3.2 illustrates the hypnograms of a MSLT recording obtained with 3 selected algorithms in comparison with the expert scoring. In general, performance of all algorithms was good capturing the naps. Performance of 4 selected methods are illustrated in Figure 3.7, and of the other methods applied in Suppl. Figures S3.11 and S3.12 and in Suppl. Tables 3.5 to 3.6. Note, that we trained only two feature-based networks with the mixture of the two datasets. Training on the mixed data resulted in an improved performance on both patient data and data of healthy participants.

3.6 Discussion

3.6.1 Comparison with human experts and automatic scoring of other groups

All implemented methods yielded reasonably high F1 scores (F1 > 0.8) for all stages when they were trained and validated on data of the same type of subjects, except for stage 1 (N1; F1<0.5). Stage 1 is known as a difficult stage to score.

F1 scores obtained with our models were comparable to the performance of human experts. Literature search showed that common measures of interrater agreement are accuracy and Cohen's Kappa (Danker-Hopfe et al., 2009, Penzel et al., 2013, Danker-Hopfe et al., 2004, Rosenberg and Van Hout, 2013). Since we used F1 scores, we can compare our results

only qualitatively with these other measures. F1 scores close to 1 correspond to good or excellent agreement and F1-score lower than 0.5 reflects poor agreement. This is similar for Cohen's Kappa and accuracy. Stage 1 was most difficult to score automatically as reflected in a low interrater agreement (Danker-Hopfe et al., 2009, Penzel et al., 2013, Danker-Hopfe et al., 2004, Rosenberg and Van Hout, 2013).

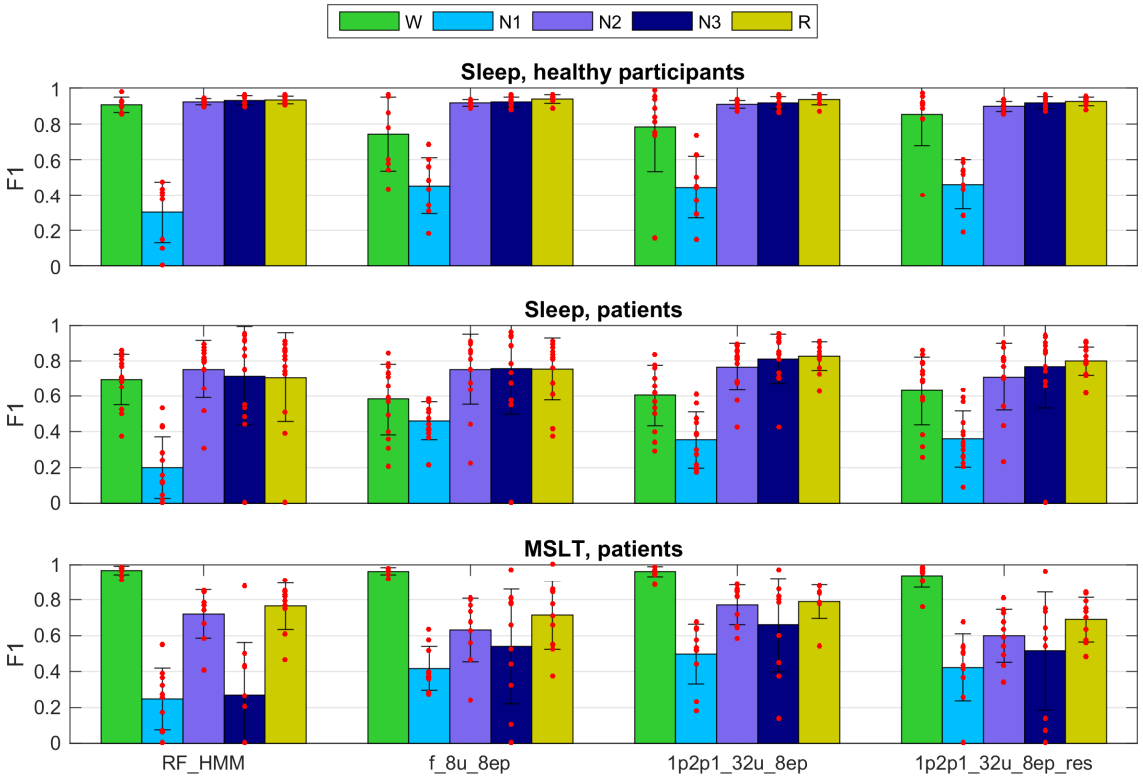


Figure 3.7. F1 scores for the methods illustrated in Figures 3.5 and 3.6 trained on a mixture of data of healthy participants and patients data (dataset 1 and 2; applied to the validation set of both datasets). Top: healthy subjects; middle: sleep recordings in patients; bottom: MSLT recordings in patients. For details see Figures 3.5 and 3.6

Performance of both LSTM and LSTM-CNN networks in our experiments were similar to the performance of recently published work where a CNN was applied to features (Tsinalis et al., 2016). We could not validate our approaches on the dataset used in the mentioned paper. They used publicly available data from PhysioNet with bipolar EEG derivations, whereas we used, as it is standard in the sleep field, EEG derivations (specifically C3A2) referenced to the contralateral mastoid.

3.6.2 Automatic scoring using different channels

Our study shows that it is possible to score sleep data with high classification accuracy using only a single EEG channel. We got slightly better results using 1 EEG, 1 EMG and 2 EOG channels.

It is difficult to conclude which method works best due to the small differences in performance. We assume that 4 channels (1 EEG, 2 EOG, 1 EMG) contain more information, but the risk of the data being noisy is also higher. This was also observed by SIESTA team (Anderer et al., 2005). The authors reported that in some cases the use of the EMG was not optimal due to a bad signal quality, and in certain cases they substituted the EMG with the high frequency content of the EEG and EOG which increased the performance of the algorithm. We also observed that a bad EMG signal reduced the performance of the algorithms.

It was surprising to observe that neural network can classify sleep, especially REM sleep, with high quality using only a single EEG channel. It is a very difficult task for a human scorer to distinguish REM based only on the EEG. Experts relie on eye movements and muscle tone (Rechtschaffen and Kales, 1968). We think that presence of patterns such as sawtooth waves

(Jouvet et al., 1960, Takahara et al., 2009) are important markers of REM sleep which helps neural network to recognize this stage.

3.6.3 Is the F1 score a good measure of scoring quality?

It is difficult to determine which method was superior based on our results. We think this is because most of our methods showed a quite good performance and produced results comparable to human experts.

Another issue is the fact that F1 scores treat epochs independently not taking the temporal structure into account and thus, we think it is not the optimal score to assess different aspects of the quality of scoring. For example, visual inspection of our results has shown that one of the problems is confusion of quiet wakefulness in the beginning of the night with REM sleep and sometimes our methods missed the first often very subtle short REM sleep episodes. Such misclassification often occurred when the EMG or EOG signals were corrupt or of bad quality. It almost does not affect F1 scores but affects the structure of sleep. In a clinical setting, such misclassifications might be intolerable as it may affect diagnosis. Thus, novel metrics to quantify the scoring quality shall be developed that take the temporal structure into account but not overestimating differences at transitions, e.g. the starting or ending of REM sleep episodes.

3.6.4 Which method is the best?

Despite the difficulties to select the best method we see some trends. Neural networks of all types detected stage 1 better than RF classifiers. This was especially evident when we applied the methods to the second dataset (patients) which indicates a better transferability of neural networks.

RF classification with HMM and MF smoothing was superior to the RF classification without smoothing, and the networks based on the raw data input tended to be superior to features based networks, in particular when they were applied to data of another lab and of a different subject population.

3.6.5 Importance of the training data

The improvement of performance was achieved when the training was performed on a mixture of the two datasets, which suggests that one should train on as diverse data as possible to reach best performance. However, the models trained only on the first dataset performed reasonably well on the second “unfamiliar” dataset showing good transferability.

In case an electrode has high impedance, the signal might become very noisy. For example, as neural nets learned that a low muscle tone is required to score REM sleep, noisy or bad EMG signals may deteriorate the performance considerably. The same holds for the EOG: if the signal quality is bad, then the algorithms may not be able to detect eye movements properly. These problems can be addressed by visual inspection of the signals before applying an algorithm and selecting the one working best with the available signals. It is also possible to develop tools for automatic examination of data quality and the subsequent selection of a corresponding algorithm.

Sometimes our models also mistakenly classified epochs close to sleep onset as REM sleep which is unlikely in healthy subjects. A human expert most likely would not make such a mistake. This can be partially explained by the fact that we never presented the whole night to our neural networks and they could thus not learn that REM sleep is unlikely to occur at the beginning of sleep. Human scorers however, have this knowledge. Some groups of patients, for example, those suffering from narcolepsy, often have REM sleep at the

sleep onset, called sleep onset REM (SOREM) sleep. Thus, it is important to be able to detect SOREM sleep episodes. They may occur also in healthy people in the early morning due to the circadian regulation of REM sleep (Mayers and Baldwin, 2005, Sharpley et al., 1996, McCarley, 2007) or by experimental manipulation (Tinguely et al., 2014). They further occur in sleep deprived subjects, and in depressed patients which are withdrawn from Selective Serotonin Reuptake Inhibitor (SSRI) medication (Mayers and Baldwin, 2005, Sharpley et al., 1996, McCarley, 2007). Therefore, we did not introduce any priors preventing our algorithm from classifying epochs at sleep onset as REM sleep.

3.6.6 Effect of the length of the sequence

We limited the length of the training sequences to 8 epochs but also tested the effect of 32 and 128-epoch long sequences. Networks trained on 128 epoch long sequences did not perform well when presented with unfamiliar datasets, i.e. they showed lower transferability. It might be that in this case the networks learned global structures of sleep and thus did not perform well on recordings with different structures (MSLT, disturbed sleep, patients, etc.). We think it is better to keep the length of the training sequence short (8 epochs).

3.6.7 Room for further improvement

We see a lot of room for further improvement. However, sleep scoring manual was mainly developed for healthy sleep, although it is being used for all different kind of patients and people under the influence of medication or drugs. Wake EEG can also be affected by substances (von Rotz et al., 2017).

Thus, we recommend extending the size of the training data including data from different laboratories, different pathologies, age groups and so on.

A major limitation of our study was the expert scoring: it was performed by single experts only. We suppose, that performance would have increased if several scorers would have scored the same data. Also, human scorers have difficulties with ambiguous data and inter-scorer variability results in part due epochs that are difficult to score with confidence (Younes et al., 2016).

We showed that our algorithms have a good generalization ability for the patient population, but the performance was not as good as on the healthy subjects. One of the possible reasons for this is the different scoring epoch length. We used the conversion procedure. It works well for the most epochs, but it is clear that there will be certain discrepancy on the borders of the stages. We think it might limit the performance, especially when these data are used for training. It was the compromise we had to make. Ideally all the data should be scored with the same epoch length.

Another aspect concerns movement time resulting in an artifact. In our datasets it was not scored, and in the AASM manual (Iber et al., 2007) scoring of movement time was abolished, which in our opinion is not optimal. Movement time basically results in EEG artifacts and it is thus difficult to assign a sleep stage. We suspect that the performance of the algorithm would improve if movements would have been scored as a separate class. Similarly, every artifact scored as some stage of sleep causes problems as artifacts do not look like sleep and thus such issues are equivalent to mistakes in the labels presented to the machine learning algorithm.

Recent work with automatic scoring on the large dataset (Sun et al., 2017) has shown that increasing the size of the dataset improved the performance. In the case of Sun et al. saturation occurred at approximately

300 recordings in the training set. However, their approach was feature based. We expect that saturation will occur at much larger numbers of recordings in the training set in case of deep neural networks working with raw data.

Raw data as an input to neural networks were recently used both with CNN (Tsinalis et al., 2016) and CNN-LSTM (Supratak et al., 2017) networks. The work of Tsinalis et al. (2016) used only a single EEG derivation did not include EOG and EMG signals and only used data of healthy participants. Our work included data from patients, different signals and we compared the performance of the different approaches. Supratak et al. (2017) included data of medicated patients (Temazepam). Both studies revealed similar performance levels as our study.

Supratak et al. (Supratak et al., 2017) used a technique known as Residual Sequence Learning, which we did not use in our models and it might improve the performance. We used residual connections in the convolutional part of the network and used different signals as independent inputs in the convolutional part of the network which were concatenated as input to the LSTM part. We think this was beneficial for the performance.

3.7 Conclusions

We demonstrated that it is possible to reliably score sleep automatically and to detect sleep onset in polysomnographic recordings using modern deep learning approaches. It was also possible to identify stage 1 and REM sleep as reliable as human experts. In general, our models provided high quality of scoring, comparable to human experts, and worked with data of different laboratories and in healthy participants and patients. Furthermore, it was possible to successfully score MSLT recordings with a different structure than night time sleep recordings. We demonstrated that temporal structure in the

data is important for sleep scoring. Some of our methods may also be applied for the on-line detection of sleep and could thus be used with mobile devices or to detect sleep in a driving simulator.

3.8 Acknowledgements

The study supported by nano-tera.ch (grant 20NA21_145929) and the Swiss National Science Foundation (grant 32003B_146643). We acknowledge NVidia for providing a GPU in the framework of academic GPU seeding.

3.9 Supplementary material

3.9.1 Definition of features

Twenty features were derived from the polysomonographic recordings (EEG, EMG, EOG). They are described in the following text. Matlab notations were used for their definitions if useful. All features were determined for consecutive 20- or 30-s epochs (epoch length used for sleep stage scoring). All signals were first resampled at 128 Hz to accommodate data recorded at different sampling rates.

The abbreviations of the features used are indicated in square brackets.

Slow waves `[slowWaves]`

We counted the number of large amplitude slow waves per 20- or 30-s epoch [#/10 s]. They were detected according to Bersagliere and Achermann (Bersagliere and Achermann, 2010). The EEG signal was band-pass filtered (pass band: 0.5-2 Hz). Half-waves were detected as negative and positive deflections between zero-crossings. We counted only the half-waves with amplitudes larger than 37.5 µV according to the scoring rules. Slow waves are

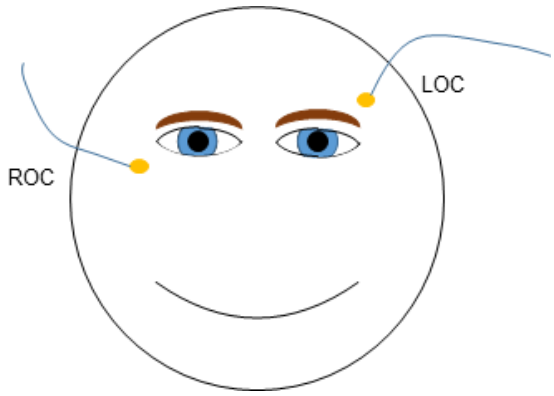
the most important marker of deep sleep. They can also be used to find REM sleep epochs due to the fact that they are absent in REM sleep (Rechtschaffen and Kales, 1968).

EMG power *[powEMG]*

To quantify the muscle tone, EMG (electromyogram) power [μV^2] in the 15-30 Hz range of consecutive 20- or 30-s epochs was determined (FFT, average of four or six 5-s epochs, Hanning window).

EOG power *[powEOG]*

We recorded two EOG (Electrooculogram) channels: one of the left (LOC) and one of the right eye (ROC). Electrodes were placed above left corner of left eye and below right corner of the right eye (Supplementary Figure S3.1). Both channels were referenced to the left mastoid (A1).



Suppl. Figure S3.1. Placement of the EOG electrodes above the outer corner of the left eye and below the outer corner of the right eye. Electrodes were referenced to the left mastoid (A1)

EOG power (1-5 Hz; [μV^2]) of the combined EOG signal (i.e. the difference between the two signals, LOC-ROC; FFT, average of four or six 5-s epochs, Hanning window) of consecutive 20 or 30-s epochs was computed.

The EOG is caused by the movement of the eyeball (Young and Sheena, 1975). The eyeball is a dipole (Du Bois-Reymond, 1848), therefore rotations of the eyeball cause changes of the electrical potentials. The electrodes were placed in such a way that eye movements cause anticorrelated changes in the two channels. That is the reason why the difference of the two channels made eye movement-related changes more prominent and reduced the noise. EOG electrodes also pick up brain activity especially during slow wave sleep. This results in the appearance of oscillations similar to eye movements during slow wave sleep. In order to prevent the confusion between eye movements and slow waves we used the ratio powEOG/Delta to capture the occurrence of eye movements (see below).

Frequency bands [Delta, Theta, Alpha, Spindles, Beta, Gamma]

EEG power [μV^2] in different frequency bands correlate with sleep stages (Aeschbach and Borbély, 1993) and thus can be used to discriminate between the different stages. For example: delta power is elevated in deep sleep (Lessard and Paschall, 1970, Borbély et al., 1981), alpha activity appears during relaxed wakefulness with closed eyes in a majority of subjects (Berger, 1929), and sleep spindles are present in stage 2 (Rechtschaffen and Kales, 1968).

We computed EEG power density spectra (FFT, average of four or six 5-s epochs, Hanning window) for consecutive 20- or 30-s epochs and determined power in the following frequency bands (in Hz):

Delta: 0.8-5.0; Theta: 5.0-8.6; Alpha: 8.6-12.0; Spindles: 11.0-15.0; Beta: 16.0-30.0; Gamma: 30.0-40.0

We also used combinations of power in those frequency bands (Louis et al., 2004):

```
(Delta.*Alpha)./(Beta.*Gamma)
```

```
Theta.^2./(Delta.*Alpha)
```

As mentioned above, powEOG/Delta was used to quantify the presence of eye movements.

Brain rate (center frequency) ***[brain_rate]***

We computed the "brain rate" [Hz] (Pop-Jordanova and Pop-Jordanov, 2005) as weighted sum of frequency values with weights equal to the relative power density in the corresponding frequency bin. It was computed in the frequency range: 0 - fs/2 Hz; fs sampling rate. Brain rate was reported to be a good measure of mental arousal (Pop-Jordanova and Pop-Jordanov, 2005). Brain rate can be computed using the BioSig package (Schlögl and Brunner, 2008).

We computed it using following Matlab code:

```
faxis = 0:df:fs/2; % frequency range
brain_rate  = (faxis*Pspec)./(sum(Pspec));
```

where Pspec is a spectrogram (matrix) with the size equal to [Number of epochs, Number of frequency bins] and df the frequency resolution (0.2 Hz in our case).

Spectral Edge Frequency (SEF) ***[SEF90, SEF50, SEFd]***

Spectral Edge Frequency (SEFxx [Hz]) (Drummond et al., 1991) is the frequency where xx percent of the power in the spectra is located below SEFxx. We abbreviate xx in percent, i.e. SEF50 denotes the frequency which divides the power density spectra in two equal parts, SEF90 the frequency which divides power density spectra in a lower part containing 90 % of the power and an upper part with 10 % of the power.

SEFxx was computed for consecutive 20- or 30-s epochs.

We used SEF50, SEF95, and their difference SEFd = SEF95-SEF50 as features. All values were computed in the frequency range of 8-16 Hz (Imtiaz and Rodriguez-Villegas, 2014).

Slow eye rolling *[SEM]*

Slow rolling eye movements occur at the transition from wake to sleep and during stage 1 (Ogilvie et al., 1988, Rechtschaffen and Kales, 1968). We implemented an algorithm developed by (Magosso et al., 2006). The method is based on a wavelet decomposition (10 levels; Daubechies wavelet of order 4 as mother wavelet). After performing the decomposition, we computed the function composed of the decomposition coefficients and thresholded them in order to detect slow eye rolling events.

We computed the amount of SEM events per 20- or 30-s epoch [#]. The input signal was computed as the difference between the LOC and ROC channels (LOC-ROC).

Eye blinks and Rapid Eye Movements (REMs) *[blinks_w, rem_w, eog_art]*

Eye blinks are important because they occur only during wakefulness. That is the reason why we expected that eye blinks would be a useful feature to discriminate between wake, stage 1 and REM sleep. Rapid eye movements occur during REM sleep; therefore, this feature would be useful to discriminate REM sleep from other stages.

We noticed that two distinct types of eye blinks exist (Supplementary Figure S3.2). In most cases we observed a strong deflection in left EOG (LOC) and only a minor anticorrelated deflection in the right EOG (ROC). In some cases we registered anticorrelated deflections of nearly equal amplitude in

both channels. We did not find information regarding these two types of eye blinks in the literature. We think that these observations can be explained in the following way: In the first case an eye blink was performed by the muscles located above the eye with only minor contraction of the muscles below the eye. In the second case a "stronger" eye blink was performed by intense contraction of muscles above and below the eye. It is important to remember that the signal resulting from eye blinks represents electrical activity of the muscles, whereas the electrical activity recorded during saccadic eye movements represent change in the electric field potential caused by rotation of the eyeballs (dipoles). The LOC channel mainly registers the activity of the muscles located above the eye and the ROC channel registers the activity of the muscles below the eye (see Supplementary Figure S3.1). Note that this asymmetric positioning of the EOG electrodes is crucial.

Eye blinks have a characteristic symmetric shape and their duration is short. We performed continuous wavelet transform with 32 levels of the LOC and ROC signals. We chose a Mexican hat wavelet because the shape of this wavelet is close to the shape of an eye blink.

```
coef_L=cwt(LOC,1:32,'mexh'); % entire night
coef_R=cwt(ROC,1:32,'mexh');
Then we sum all the coefficients and get two signals

wl = sum(coef_L(:,:)); % entire night
wr = sum(coef_R(:,:));
```

wl, wr are vectors (samples). The next step was to find peaks in wl (corresponding to LOC). We selected peaks with minimal height of 4000 separated by at least 0.2 s. (Note that function findpeaks was introduced recently in Matlab; we used version 2015b).

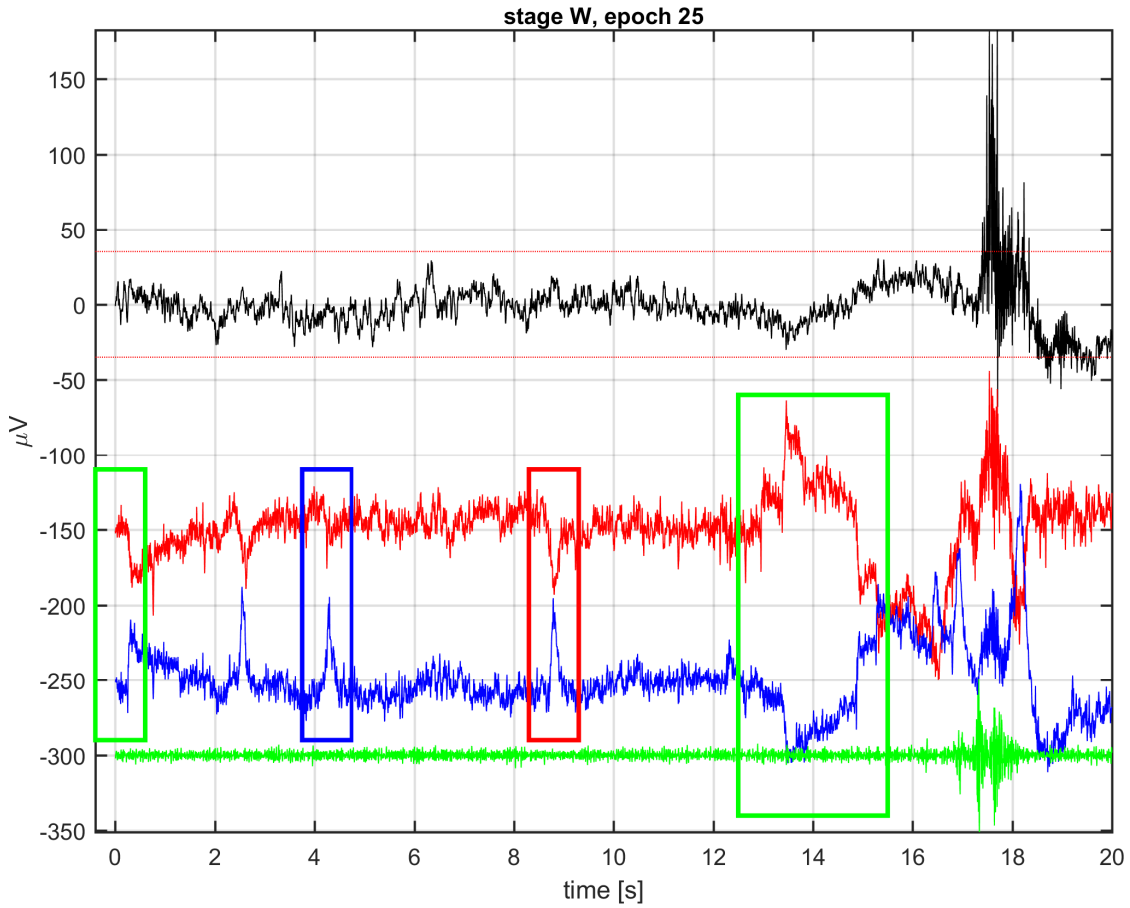
```
[wl_peak_amp, wl_peak_pos, wl_peak_widths] = findpeaks(wl, 'MinPeakHeight',4000, 'MinPeakDistance', round(0.2*fs));
% fs: sampling rate
```

Then we selected only the peaks with an amplitude ratio wl/wr smaller than -2. This condition ensures that we reject positively correlated deflections and requires that at least a minor anticorrelated deflection in ROC is present, which is usually the case for eye blinks. A second condition was the following – the ratio of the amplitude to the width of the peak should be > 150 samples (approx. 1 s) as we only need to consider narrow peaks because eye blinks are short lasting events.

Following Matlab code implements two conditions mentioned above:

```
ndx1 = wl_peak_pos(find(wl(wl_peak_pos)./wr(wl_peak_pos)<-2.0));
ndx2 = wl_peak_pos(find((wl(wl_peak_pos)./wl_peak_widths)>150));
blnkpos = intersect(ndx1, ndx2);
```

The sum of detected eye blink events in consecutive 20- or 30-s epochs [#] forms a feature for the classifier (blinks_w).



Suppl. Figure S2. Example of one 20-s epoch of wakefulness. Black: EEG derivation C3A2 (red lines indicate ± 37.5 µV); red: right EOG (ROC, Suppl. Fig. S1); blue: left EOG (LOC); green: chin EMG. Two types of eye blinks (blue and red rectangles) and a saccadic eye movement (green rectangle) are illustrated. Moreover, between 15 and 20 s a movement artifact occurred affecting all channels

A next important step is to detect ***rapid eye movements*** and saccadic eye movements. These events have similar amplitude as eye blinks (Supplementary Figure S2; green rectangle). They are characterized by a very steep initial deflection followed by a slow recovery. The deflections in LOC and ROC are strongly anticorrelated and have similar amplitudes. Since there is a step-like change we used a Haar wavelet to capture it.

We performed continuous wavelet transform (Haar wavelet) with 32 levels of the LOC and ROC signals and summed up of the coefficients.

```
coef_L=cwt(LOC,1:32,'haar');
coef_R=cwt(ROC,1:32,'haar');
wl = sum(coef_L(:,:));
wr = sum(coef_R(:,:));
```

We also computed the correlation between LOC and ROC (Cor) on a sliding window. Window length was 1/8 s; moving step 1 sample.

Our main signal was a multiplication of wl, wr and (1-Cor) with normalizing constant of 1/200000. This signal is high when both wl and wr are high and anticorrelated.

Following Matlab code performs the computation:

```
wlwr = wl.*wr/200000.*(Cor'-1);
```

Then we select the peaks in wlwr:

```
[wlwr_peak_amp, peak_pos, peak_widths1 ] = findpeaks(
wlwr, 'MinPeakHeight',10, 'MinPeakDistance',
round(0.05*fs));
```

We required a minimal height of 10 and minimal distance between peaks of 0.05 s. Saccadic eye movements can occur very quickly one after another. That is the reason we have chosen such a short interval.

The next step was to filter out the peaks which correspond to rapid and saccadic eye movements.

We required a ratio of amplitudes between wl and wr of -0.3 and -1.7. Ideally it should be equal to -1 but, it may vary depending on the electrode position and signal quality.

Moreover, the ratio of the amplitude of wlwr to the width of the peak should be >1 to make sure that we do not confuse REMs with artifacts and slow eye movements. We also constrained the width of the peak. It had to be wider than 5 samples. This is very short but filters out artifacts (spikes).

```
ndx11 = peak_pos(find(wl(peak_pos)./wr(peak_pos)<-0.3));
ndx12 = PKpos(find(wl(peak_pos)./wr(peak_pos)>-1.7));
ndx1 = intersect(ndx11, ndx12);
ndx2 = peak_pos(find((wlwr(peak_pos)./peak_widths1)>1.0));
ndx3 = peak_pos(find(peak_widths1>5));

ndx23 = intersect(ndx2, ndx3);
rempos = intersect(ndx1, ndx23);
```

The number of detected REMs in consecutive 20- or 30-s epochs [#] formed a feature (REM_w).

Note that the thresholds for these two methods were derived based on our experience and common sense. We think that quantitative adjustment of the thresholds might improve the performance of eye movement detection. Note that the choice of the wavelets is a very strong prior and a crucial parameter of the methods.

We also used a feature ***EOG artifact*** (eog_art) in order to detect epochs where EOG channels were contaminated with artifacts. This feature is the amount of samples in LOC and ROC exceeding an absolute value of 350 μV.

Feature vector

The final feature vector (20 components) is composed of the above defined features:

```
features_names = {'slowWaves', 'EMG',
'EOG/Delta','Spindles', 'Delta', ...
'Theta', 'Alpha', 'Beta', 'Gamma', ...
'Alpha/Theta', 'Beta/Theta', 'Alpha/Delta', 'Delta/Theta',
...
'(Delta*Alpha)/(Beta*Gamma)', 'Theta^2/(Delta*Alpha)',
'Brain rate', ...
'blinks_wav', 'rem_wav', 'SEM', 'eog_art'};

features = [slowWaves, powEMG, powEOG./Delta, Spindles,
Delta, Theta, ...
     Alpha, Beta, Gamma, Alpha./Theta, Beta./Theta,
Alpha./Delta, ...
     Delta./Theta, (Delta.*Alpha)./(Beta.*Gamma),
Theta.^2./(Delta.*Alpha),...
     Brain_rate', blinks_w', rem_w', SEM', eog_art' ];
```

Following transformation was applied:

```
ndx = [2:16];
features = log(log(features +1)+1);
```

```
features (:,ndx) = log(features (:,ndx)+1);
```

The original data were extremely skewed and we reduced the skewness by this transformation. We did not expect that a monotonous transformation would affect the random forest algorithm, but it might affect artificial neural networks.

3.9.2 Taking the temporal structure into account by a Hidden Markov Model (HMM)

This model assumes that the system has a hidden state. This hidden state changes with certain probabilities, which are called transition probabilities. Transition of the system into the new state depends only on the current state (that's why it is called Markov model), i.e. the system has memory only of one step (20- or 30-s epoch). The matrix of transition probabilities is called a transition matrix. As we cannot observe the hidden state directly, that is why it is referred to as a hidden state. Instead we can measure an observable variable. In our case the hidden state would be the sleep stage, whereas the observable variable is the probability vector resulting from the RF algorithm. We computed the transition matrix from the training set and applied the Viterbi algorithm (Viterbi, 1967) to infer the most probable sequence of stages. We employed RF and the Viterbi algorithm implementations of MATLAB version 2015a. We assumed uniform prior probabilities of all classes for RF meaning that without any information about the epoch the classifier would predict any of the classes with equal probability.

3.9.3 Optimization

Gradient descent

One of the well-known algorithms of optimization is the gradient descent (GD). GD is based on the idea that if one moves along the maximal gradient in small steps one will end up in a minimum of a function.

Gradients are easy to compute for analytical functions. However, they are not easy to compute for the cost function of neural networks. Thus, special algorithms are used. The most widely applied algorithm for computing weight gradients of the neural networks is backpropagation (Werbos, 1974, Werbos, 1994). After the gradients are computed they are used to adjusts the weights accordingly (Bishop, 2016).

The problem of the gradient descent algorithm is that it requires the whole dataset to calculate the gradients. There are some tricks which help to reach the minimum faster. The algorithm still can converge to the optimum even if we use only one randomly selected training example from the training set to compute the gradients. The convergence happens much faster. On the other hand, such methods result in large fluctuations of the gradients. This method is called stochastic gradient descent (Bishop, 2016).

Usually gradients are computed over several data points to reduce fluctuations. These sets of data points are called batches. First, we need to split the whole training set into batches (see above).

When we have gone through all batches we have accomplished one training iteration (usually it is called training epoch, but we call it iteration to avoid confusion with the scoring epochs of the EEG data).

Another trick is to carry over some gradient from previous steps, i.e. to add momentum (Sutskever et al., 2013). It helps to reduce fluctuations of the gradient. One can imagine it with a very simply analogy: When you ski down

the mountain you don't change your direction on every small bump, you have a momentum directed towards the valley. There are several different ways to add the momentum to a gradient.

We trained the networks using the Adam (Adaptive moment estimation) (Kingma and Ba, 2014) algorithm with Nesterov momentum (Nesterov, 1983).

We also clipped the gradients: its norm could not be larger than 1. We used clipping only for the raw data based networks. Gradient clipping prevents gradients from becoming too large. If gradients become too large the convergence usually does not occur. This phenomenon is called explosion of gradients.

Regularization

Regularization is needed to prevent overfitting. Different approaches might be applied. The simplest one is to penalize the weights. The most common approach used for neural networks is the dropout regularization.

We used both recurrent and non-recurrent dropouts (Hinton et al., 2012, Srivastava et al., 2014). Dropout regularization switches off certain neurons during training. It is considered to be an efficient regularization method for neural networks (Srivastava et al., 2014). The value of both types of dropouts in our networks was equal to 0.25. It means that 25% of the neurons were randomly switched off at each iteration.

The number of epochs of all classes (sleep stages) in the data is unequal. Moreover, the distribution of epochs of the different classes might differ between the training and test data. Thus, we assigned a weight to every class. In this way, every class contributed equally to the loss function as if there were equal amounts of epochs of all classes in the training data. The weight of a class X was equal to the ratio of the frequency of the most frequent class (in

our case S2) to the frequency of class X. Frequencies and weights were computed within a batch.

3.9.4 Batches

Our batches consisted of a specific number of sequences, each of them 8, 32 or 128 epochs long. The number of sequences in a batch was adapted to keep amount of data per batch similar for different sequence lengths.

For both types of networks (features and raw data) we applied the following parameters (note that in this context a sample is a sequence of scoring epochs):

samples_per_batch – number of sequences in one batch
samples – number of sequences sampled from each recording from sample_files
sample_files - number of files of the training set randomly chosen for every training iteration.

Thus, number of batches in every training iteration was `(sample_files*samples)//samples_per_batch.` "//" means integer division.

For training of the LSTM models on every training iteration we sampled (512 or 32) sequences with the corresponding length out of each recording. Samples for each batch were chosen randomly out of this subset. We chose `sample_files = 36; samples_per_batch=512 or 32; samples = samples_per_batch.`

For the CNN-LSTM models we had following numbers:

8-epoch long sequences: `samples_per_batch = 100`

32-epoch long sequences: `samples_per_batch = 40`

128 epoch long sequences: `samples_per_batch = 10`

Sample_files was 16 for and samples was set to 200 for all cases except the network with 128 epoch long sequences. For latter network we set Samples = 100 due to the memory restrictions. Note that we used a random subset of the training data in each training iteration.

Thus, on every training iteration we randomly chose 16 recordings (sample_files) from the training set and sampled 200 or 100 sequences (samples) from each recording. Even though each training iteration did not contain all the training data (only 16 recordings), overall the networks were trained using the complete training set.

Even though we kept the amount of epochs per batch constant, the overall number of epochs per training iteration was proportional to the length of the input sequence. This might be a limitation of our study because networks trained with longer sequences would overfit earlier than the ones with shorter sequences.

3.9.5 Training and validation

We used three machine learning approaches: random forests (RF) based on features, feature based networks (LSTM) and raw-data based networks (CNN-LSTM). We first trained all algorithms on the dataset 1 comprised of healthy participants (36 recordings) and validated them on the validation part (9 recordings) and test part (9 recordings) and on dataset 2 (patients, 43 recordings). In the next step, we trained all models using a mixture of the two datasets (55 recordings: 36 healthy sleepers and 19 patients) and validated on

the mixed validation set (33 recordings: 9 nights of healthy subjects, 14 nights of sleep in patients and 10 MSLT recordings of patients). The idea was to test whether our models are transferable to datasets from another laboratory and to a different subject population (patients).

A difficulty in using a combination of both datasets for CNNs was the fact that sleep stage scoring was performed with a different epoch length (20 and 30 s) in the two datasets. We overcame this problem by converting the labels of the second dataset scored with 30-s epochs. We represented every 30-s epoch as three dummy 10-s epochs, all of them having identical labels. Then we reorganized the whole night in sequences of 20-s epochs consisting of two dummy epochs. Every such 20-s epoch was labeled according to the last (second) dummy epoch of this 20-s epoch.

3.9.6 Naming conventions of algorithms

RF classification

RF stands for Random Forest, RF_HMM means that the classification was smoothed using a HMM (see above), and RF_MF indicates smoothing with a moving median filter of length 3 (three 20- or 30-s epochs).

LSTM networks

The structure of the networks was encoded in the name <f/s>_<un>u_<en>ep: <f/s> specifies the input of the neuronal network, with ‘f’ features and ‘s’ spectrograms of the EEG. <un> reflects the amount of LSTM units per layer (always 3 layers). <en> codes the length of the sequence used for training. If ‘1_dir’ follows at the end, it indicates that the network was unidirectional, otherwise it was bidirectional, i.e. it had information from the future for classification.

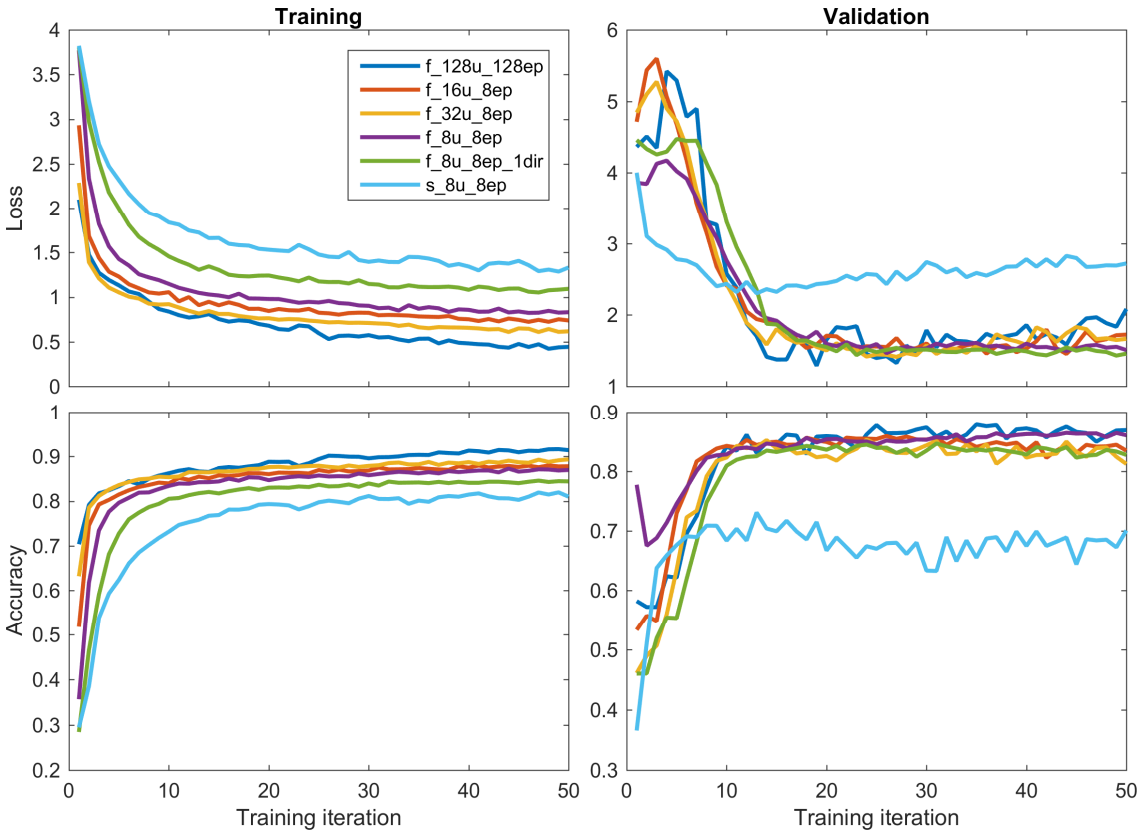
For example, *f_16u_8ep* means features as input, 16 units in each of the 3 layers, a sequence length of 8 epochs was used for training, and it was a bidirectional network. Recurrent activation functions of the LSTM were sigmoid and activation functions of LSTM were tanh.

CNN-LSTM networks

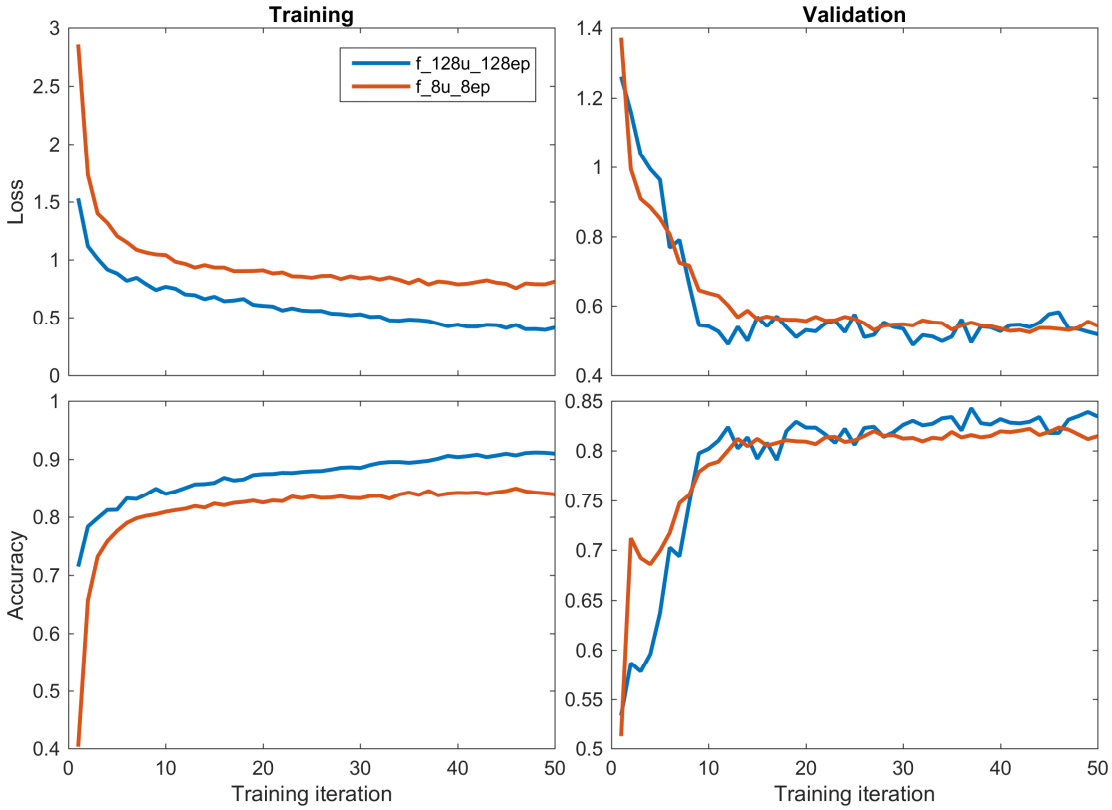
The structure of the network was encoded in the name <input>_<un>u_<en>ep: <input> can be "1p" – a single raw EEG channel as input; "1p2" – a raw EEG and two raw EOG channels as input; "1p2p1" – same as previously and additionally EMG (muscle tone) as input; "p" stands for plus. <un> indicates the number of LSTM units per layer (always 2 layers). <en> codes the length of the sequence used for training. If 'res' follows at the end, it indicates that the network had residual connections.

For example, *1p2p1_32u_8ep_res* means raw EEG, EOG and muscle tone as input, 32 LSTM units per layer, 8 epoch sequence length for training, and residual connections. Recurrent activation functions of the LSTM were sigmoid and activation functions of LSTM were tanh. Activation function of all convolutional layers was ReLU.

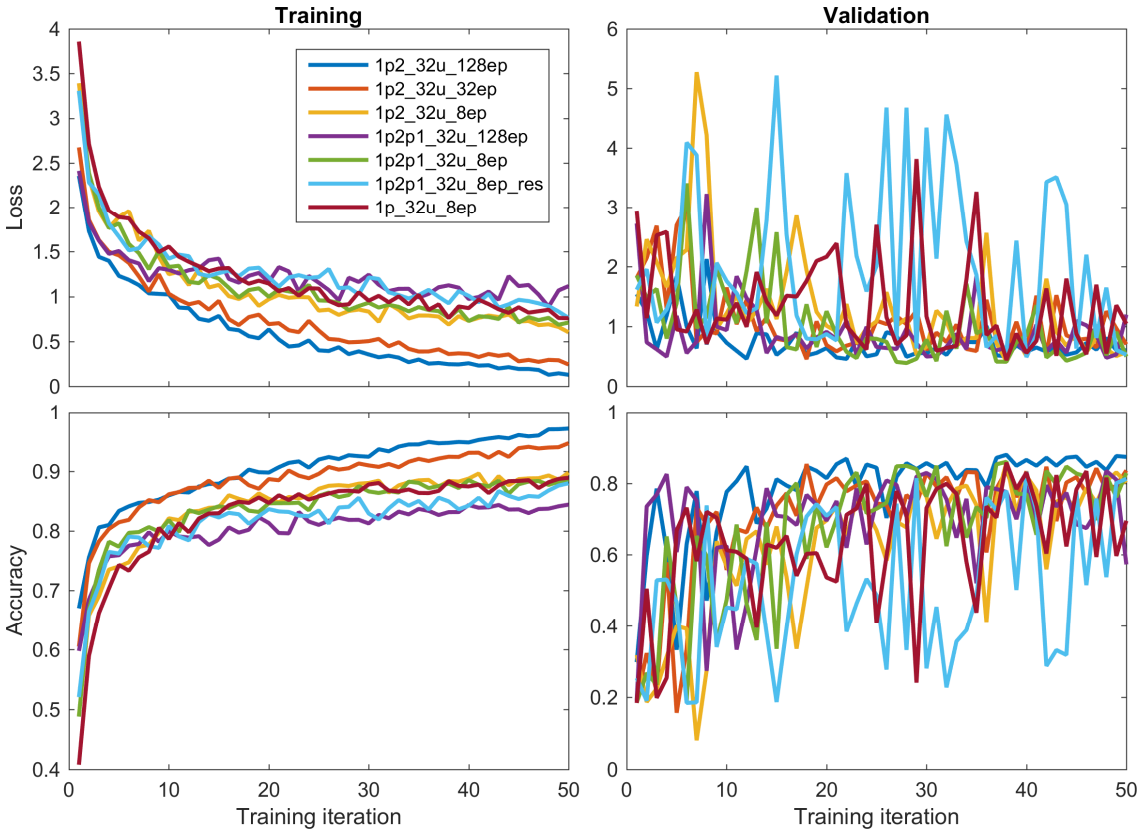
3.9.7 Training and validation



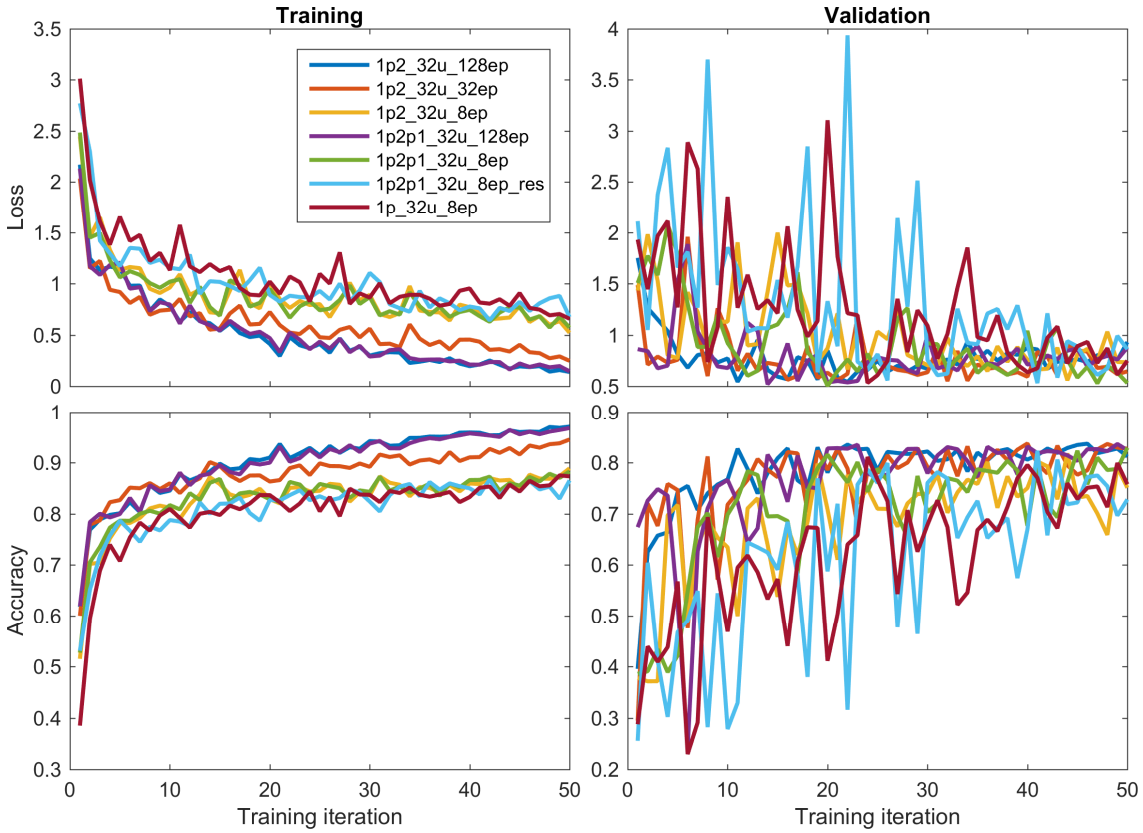
Suppl. Figure S3.3. Learning curves of LSTM neuronal networks trained on dataset 1 (healthy participants). The networks were trained for 50 epochs (iterations, these epochs are not related to scoring epochs). The structure of the network was encoded in the name (see supplementary material for the naming convention). Left, loss and accuracy computed on the training data, right: on the validation data



Suppl. Figure S3.4. Learning curves of LSTM neuronal networks trained on a mixture of healthy subjects and patients (datasets 1 and 2). For details see Suppl. Figure S3.3. Only 2 networks were trained

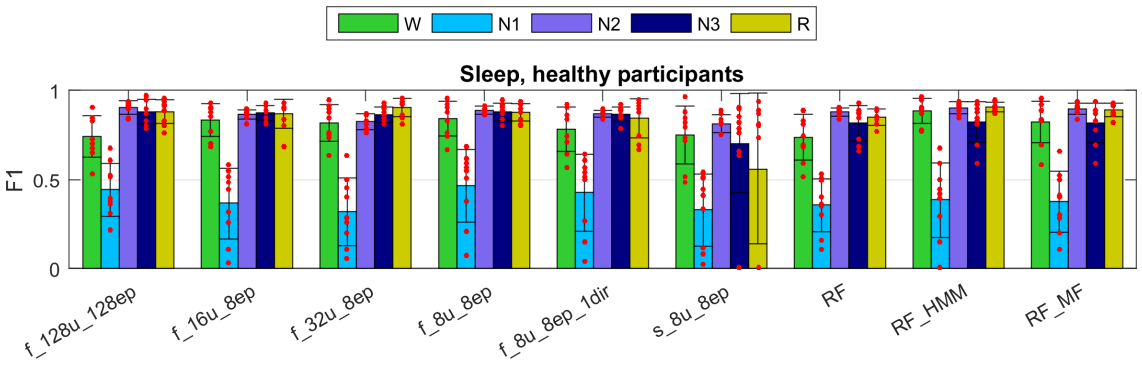


Suppl. Figure S3.5. Learning curves of CNN_LSTM neuronal networks with raw data as input, trained on dataset 1 (healthy participants). For details see Suppl. Figure S3.3



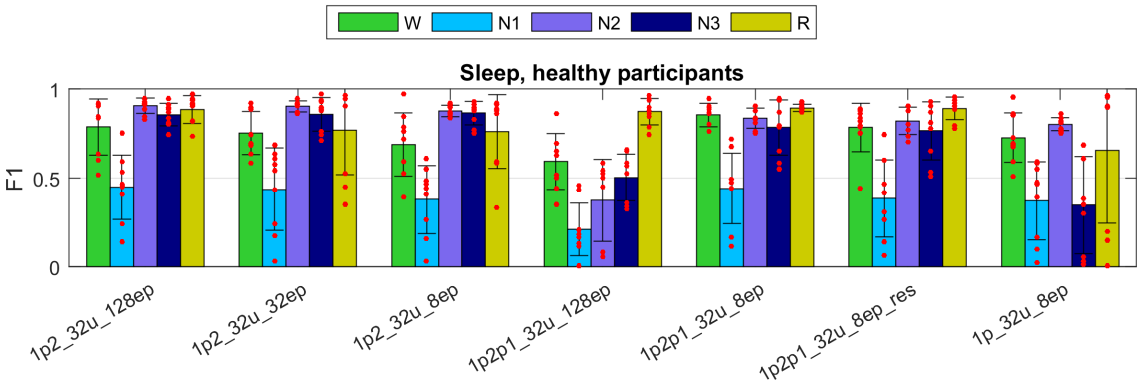
Suppl. Figure S3.6. Learning curves of CNN_LSTM neuronal networks with raw data as input, trained on a mixture of healthy subjects and patients (datasets 1 and 2). For details see Suppl. Figure S3.3

3.9.8 Performance evaluation



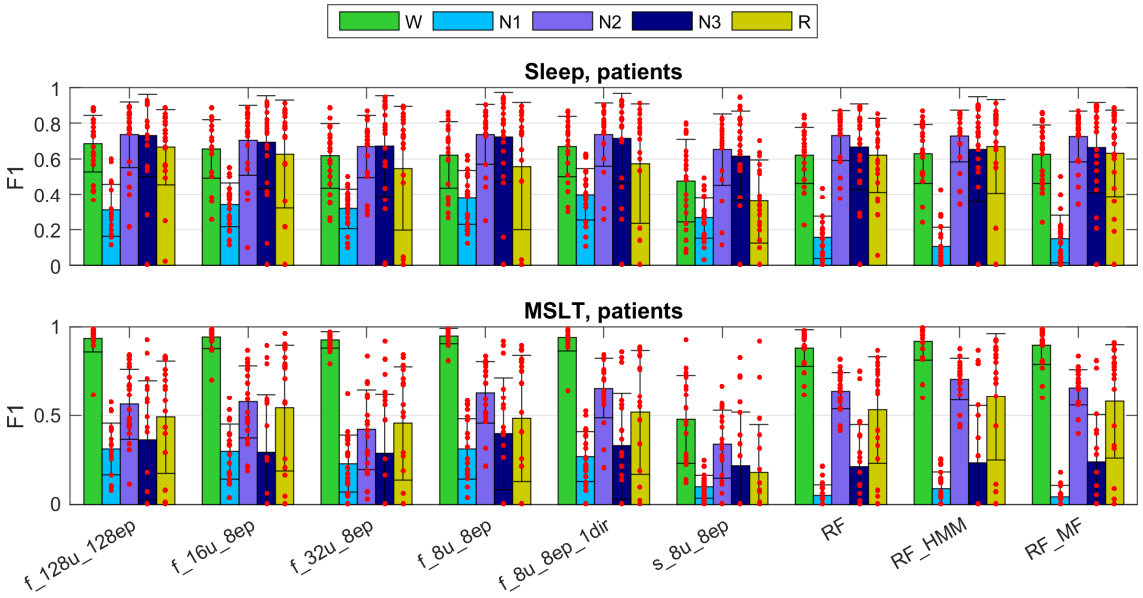
Suppl. Figure S3.7. F1-scores obtained with LSTM networks and RF classifiers and features as input. The algorithms were applied to the validation set (9 recordings of dataset 1) of healthy subjects. The first 6 groups of bars represent various neuronal networks. Values for RF classifiers are shown for comparison.See text in supplementary material for the naming conventions of the classifiers. Mean ± SD are shown; red dots represent F1 values of single recordings. Feature vectors were computed for consecutive 20-s epochs and contained 20 features

For the exact performance on the validation and the test set see Suppl. Table 3.1.

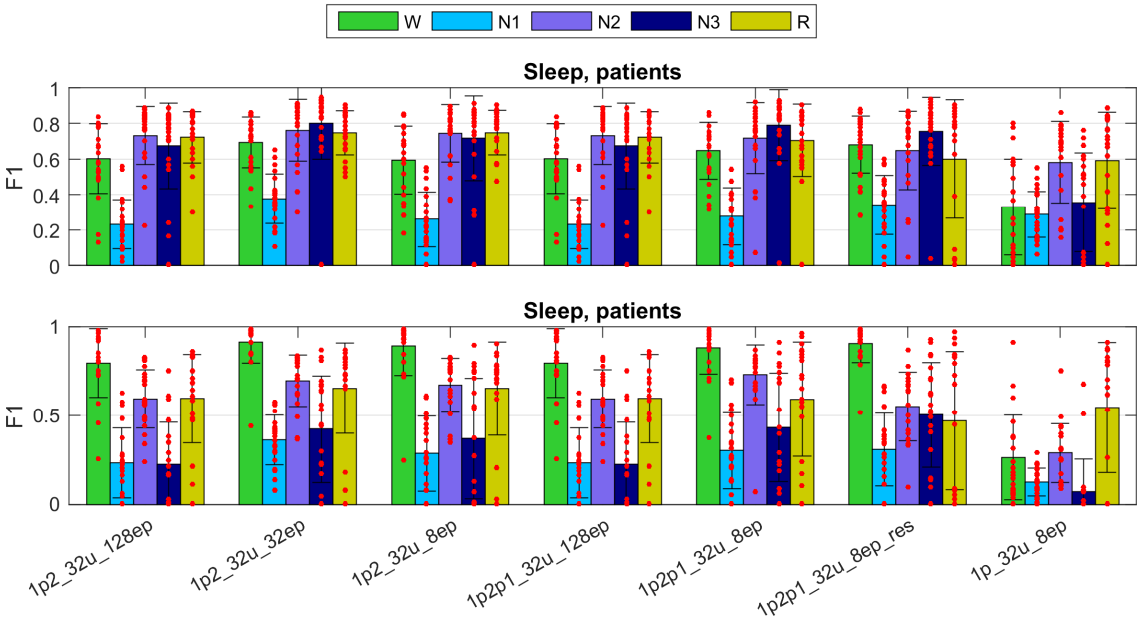


Suppl. Figure S3.8. F1 scores obtained with CNN-LSTM networks and raw data as input. The algorithms were applied to the validation set of dataset 1 (healthy subjects). See text in supplementary material for the naming conventions of the classifiers and Suppl. Figure S3.7 for further details

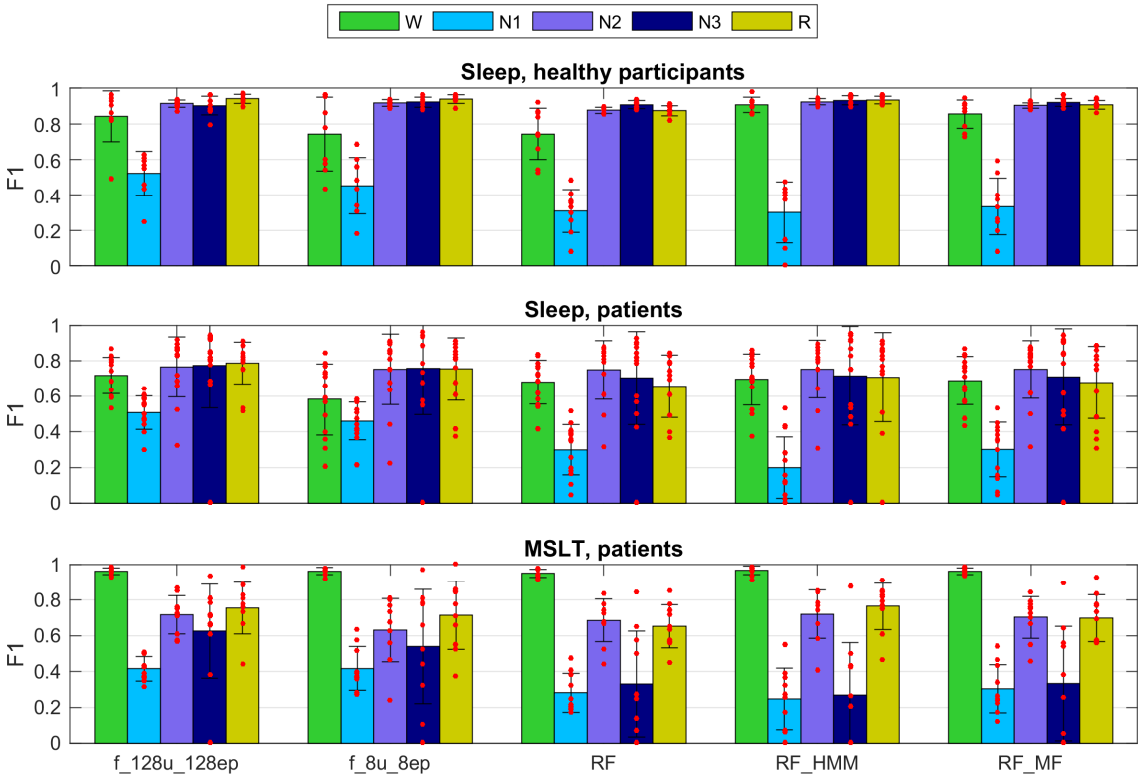
For the exact performance on the validation and the test set see Suppl. Table 3.2.



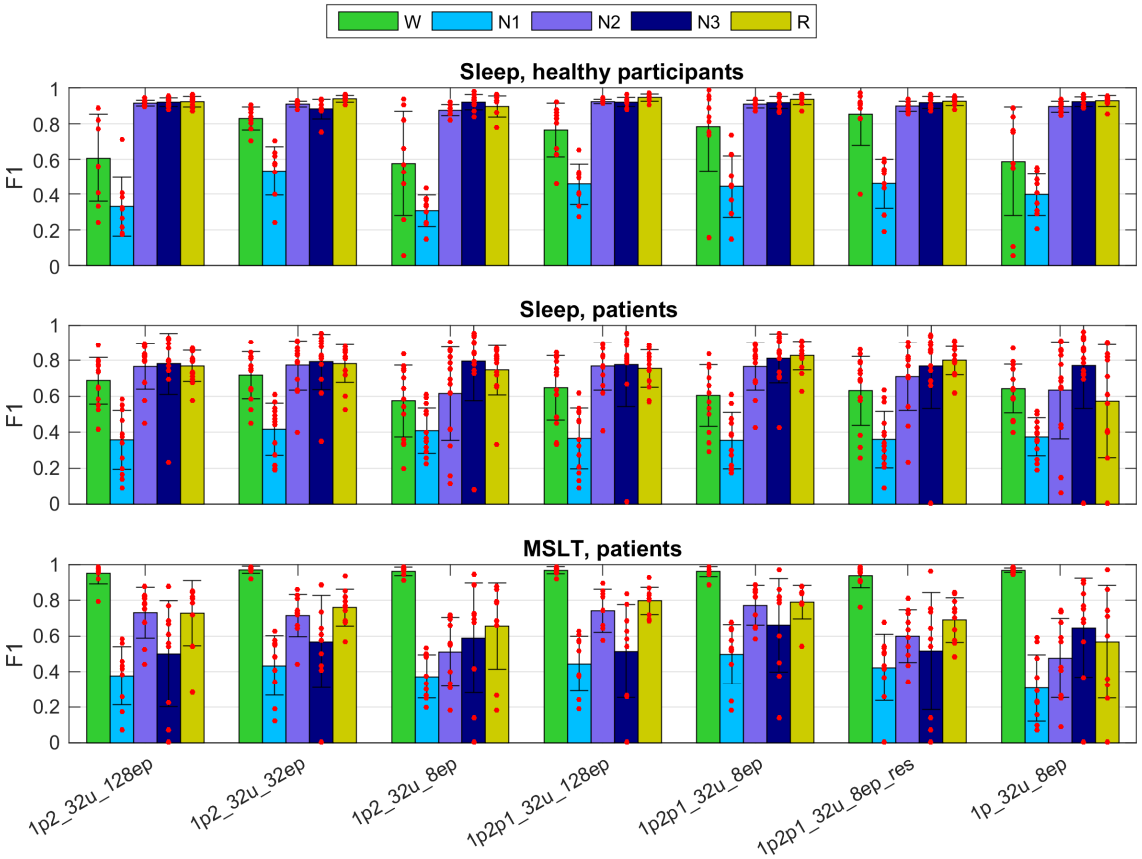
Suppl. Figure S3.9. F1-scores obtained with LSTM networks and RF classifiers trained on the healthy subjects applied to patient data (dataset 2). The classifiers were the same as in Suppl. Figure S3.7. ***Top:*** night sleep recordings; ***bottom:*** MSLT recordings. Note that some MSLT recordings did not contain any stage 3 epochs. Such recordings were not taken into account in the computation of the average F1 score and the standard deviation for stage 3 (N3). For further details see Suppl. Figure S3.7. For the exact performance see Suppl. Table 3.3.



Suppl. Figure S3.10. F1 scores obtained with CNN-LSTM networks and raw data as input trained on the healthy subjects and applied to the patient data. ***Top:*** sleep recordings; ***bottom:*** MSLT recordings. Note that some MSLT recordings did not contain any stage 3 epochs. These recordings were not considered in the computation of the average F1 score and standard deviation for stage 3 (N3). See text in supplementary material for the naming conventions of the classifiers and Suppl. Figure S3.7 for further details.
For the exact performance see Suppl. Table 3.4.



Suppl. Figure S3.11. F1-scores obtained with LSTM networks and RF classifiers trained on a mixture of data of patients and healthy participants. See text in supplementary material for the naming conventions of the classifiers. ***Top:*** results of combined validation and test set (18 recordings) of dataset 1; ***middle:*** results of sleep recordings of test data of dataset 2 (patients); ***bottom:*** results of MSLT recordings of test data of dataset 2 (patients). For the exact performance see Suppl. Table 3.5.



Suppl. Figure S3.12. F1-scores obtained with CNN-LSTM networks and raw data as input trained on a mixture of data of patients and healthy participants. ***Top:*** results of combined validation and test set (18 recordings) of dataset 1; ***middle:*** results of sleep recordings of test data of dataset 2 (patients); ***bottom:*** results of MSLT recordings of test data of dataset 2 (patients). See text in supplementary material for the naming conventions of the classifiers and Suppl. Figure S3.7 for further details. For the exact performance see Suppl. Table 3.6.

	Validation				
	W	1	2	3	R
f_128u_128ep	0.74 (0.12)	0.44 (0.15)	0.90 (0.04)	0.88 (0.07)	0.88 (0.07)
f_16u_8ep	0.83 (0.09)	0.36 (0.20)	0.86 (0.03)	0.87 (0.04)	0.87 (0.08)
f_32u_8ep	0.82 (0.10)	0.32 (0.19)	0.82 (0.04)	0.86 (0.04)	0.90 (0.05)
f_8u_8ep	0.84 (0.10)	0.46 (0.21)	0.89 (0.02)	0.88 (0.05)	0.88 (0.05)
f_8u_8ep_1dir	0.78 (0.12)	0.43 (0.22)	0.87 (0.02)	0.87 (0.04)	0.84 (0.11)
s_8u_8ep	0.75 (0.16)	0.33 (0.21)	0.81 (0.05)	0.70 (0.28)	0.56 (0.42)
RF	0.74 (0.13)	0.35 (0.15)	0.88 (0.02)	0.82 (0.10)	0.85 (0.05)
RF_HMM	0.88 (0.07)	0.38 (0.21)	0.90 (0.03)	0.82 (0.11)	0.91 (0.03)
RF_MF	0.82 (0.12)	0.37 (0.17)	0.90 (0.03)	0.82 (0.11)	0.89 (0.04)
	Test				
	W	1	2	3	R
f_128u_128ep	0.87 (0.08)	0.49 (0.13)	0.91 (0.02)	0.92 (0.04)	0.94 (0.03)
f_16u_8ep	0.93 (0.01)	0.50 (0.13)	0.91 (0.01)	0.94 (0.02)	0.95 (0.03)
f_32u_8ep	0.91 (0.05)	0.41 (0.14)	0.91 (0.01)	0.93 (0.02)	0.92 (0.03)
f_8u_8ep	0.92 (0.04)	0.49 (0.14)	0.92 (0.01)	0.94 (0.02)	0.94 (0.03)
f_8u_8ep_1dir	0.86 (0.09)	0.44 (0.10)	0.90 (0.02)	0.93 (0.02)	0.92 (0.03)
s_8u_8ep	0.77 (0.17)	0.24 (0.08)	0.85 (0.04)	0.85 (0.07)	0.61 (0.14)
RF	0.74 (0.13)	0.35 (0.15)	0.88 (0.02)	0.82 (0.10)	0.85 (0.05)
RF_HMM	0.91 (0.04)	0.33 (0.16)	0.92 (0.02)	0.93 (0.03)	0.92 (0.04)
RF_MF	0.84 (0.11)	0.32 (0.17)	0.90 (0.01)	0.92 (0.02)	0.90 (0.04)

Suppl. Table 3.1. F1-scores of the feature-based algorithms on the validation (top part) and test set (bottom part) of dataset 1 (healthy participants). Mean values (standard deviations) are shown. See supplementary material for the naming of the algorithms. W: waking; 1 - 3: NREM sleep stages; R: REM sleep.

	Validation				
	W	1	2	3	R
1p2_32u_128ep	0.79 (0.16)	0.45 (0.18)	0.91 (0.04)	0.86 (0.06)	0.88 (0.08)
1p2_32u_32ep	0.75 (0.12)	0.44 (0.23)	0.90 (0.03)	0.86 (0.09)	0.77 (0.25)
1p2_32u_8ep	0.69 (0.18)	0.38 (0.19)	0.88 (0.03)	0.86 (0.07)	0.76 (0.21)
1p2p1_32u_128ep	0.59 (0.16)	0.21 (0.15)	0.37 (0.23)	0.50 (0.13)	0.87 (0.07)
1p2p1_32u_8ep	0.85 (0.07)	0.44 (0.20)	0.84 (0.06)	0.78 (0.16)	0.89 (0.02)
1p2p1_32u_8ep_res	0.78 (0.14)	0.38 (0.22)	0.82 (0.08)	0.77 (0.16)	0.89 (0.06)
1p_32u_8ep	0.73 (0.14)	0.37 (0.22)	0.80 (0.04)	0.35 (0.27)	0.66 (0.41)
	Test				
	W	1	2	3	R
1p2_32u_128ep	0.87 (0.08)	0.52 (0.12)	0.92 (0.02)	0.91 (0.03)	0.94 (0.02)
1p2_32u_32ep	0.86 (0.11)	0.51 (0.15)	0.92 (0.02)	0.92 (0.04)	0.95 (0.03)
1p2_32u_8ep	0.81 (0.24)	0.50 (0.11)	0.92 (0.01)	0.93 (0.02)	0.94 (0.03)
1p2p1_32u_128ep	0.65 (0.30)	0.20 (0.22)	0.32 (0.17)	0.53 (0.08)	0.89 (0.05)
1p2p1_32u_8ep	0.86 (0.08)	0.50 (0.19)	0.89 (0.05)	0.88 (0.06)	0.94 (0.03)
1p2p1_32u_8ep_res	0.89 (0.04)	0.35 (0.21)	0.84 (0.09)	0.84 (0.09)	0.93 (0.04)
1p_32u_8ep	0.83 (0.16)	0.44 (0.10)	0.84 (0.03)	0.69 (0.09)	0.92 (0.03)

Suppl. Table 3.2. F1-scores of the raw data based algorithms on the validation (top part) and test set (bottom part) of dataset 1 (healthy participants). Mean values (standard deviations) are shown. See supplementary material for the naming of the algorithms. W: waking; 1 - 3: NREM sleep stages; R: REM sleep.

	Sleep patients				
	W	1	2	3	R
f_128u_128ep	0.69 (0.16)	0.31 (0.15)	0.74 (0.18)	0.73 (0.23)	0.67 (0.21)
f_16u_8ep	0.66 (0.16)	0.34 (0.13)	0.70 (0.20)	0.69 (0.26)	0.63 (0.31)
f_32u_8ep	0.62 (0.18)	0.32 (0.11)	0.67 (0.18)	0.67 (0.28)	0.55 (0.35)
f_8u_8ep	0.62 (0.19)	0.38 (0.15)	0.74 (0.17)	0.72 (0.25)	0.56 (0.36)
f_8u_8ep_1dir	0.67 (0.17)	0.40 (0.15)	0.74 (0.18)	0.71 (0.25)	0.57 (0.34)
s_8u_8ep	0.48 (0.23)	0.27 (0.12)	0.65 (0.20)	0.62 (0.25)	0.36 (0.24)
RF	0.62 (0.16)	0.16 (0.12)	0.73 (0.14)	0.67 (0.24)	0.62 (0.21)
RF_HMM	0.63 (0.17)	0.10 (0.11)	0.73 (0.15)	0.65 (0.30)	0.67 (0.26)
RF_MF	0.63 (0.16)	0.15 (0.13)	0.73 (0.14)	0.66 (0.25)	0.63 (0.24)
	MSLT patients				
	W	1	2	3	R
f_128u_128ep	0.94 (0.08)	0.31 (0.15)	0.56 (0.20)	0.36 (0.34)	0.49 (0.32)
f_16u_8ep	0.94 (0.06)	0.30 (0.16)	0.58 (0.20)	0.29 (0.33)	0.54 (0.35)
f_32u_8ep	0.93 (0.05)	0.23 (0.16)	0.42 (0.23)	0.29 (0.34)	0.46 (0.32)
f_8u_8ep	0.95 (0.04)	0.31 (0.17)	0.63 (0.17)	0.40 (0.32)	0.48 (0.36)
f_8u_8ep_1dir	0.94 (0.07)	0.27 (0.14)	0.66 (0.17)	0.33 (0.30)	0.52 (0.35)
s_8u_8ep	0.48 (0.25)	0.10 (0.06)	0.34 (0.19)	0.22 (0.30)	0.18 (0.27)
RF	0.88 (0.10)	0.05 (0.06)	0.64 (0.10)	0.21 (0.24)	0.53 (0.30)
RF_HMM	0.92 (0.11)	0.09 (0.09)	0.71 (0.12)	0.23 (0.32)	0.61 (0.36)
RF_MF	0.90 (0.11)	0.04 (0.06)	0.66 (0.10)	0.24 (0.27)	0.58 (0.32)

Suppl. Table 3.3. This table represents F1 values of the feature-based algorithms which were trained on the subjects dataset, i.e. the same models as in the Suppl. Table 3.1, but they were validated on the patient dataset. Sleep recordings (top part) and MSLT recordings (bottom part) were analyzed separately. Standard deviations are shown in bracket. For the meaning of the short names see the corresponding figure.

	Sleep patients				
	W	1	2	3	R
1p2_32u_128ep	0.60 (0.20)	0.23 (0.14)	0.73 (0.16)	0.67 (0.24)	0.72 (0.14)
1p2_32u_32ep	0.69 (0.14)	0.38 (0.14)	0.76 (0.17)	0.80 (0.20)	0.75 (0.12)
1p2_32u_8ep	0.59 (0.19)	0.26 (0.15)	0.75 (0.16)	0.72 (0.24)	0.75 (0.13)
1p2p1_32u_128ep	0.60 (0.20)	0.23 (0.14)	0.73 (0.16)	0.67 (0.24)	0.72 (0.14)
1p2p1_32u_8ep	0.65 (0.16)	0.28 (0.16)	0.72 (0.20)	0.79 (0.20)	0.70 (0.20)
1p2p1_32u_8ep_res	0.68 (0.16)	0.34 (0.17)	0.65 (0.22)	0.76 (0.19)	0.60 (0.33)
1p_32u_8ep	0.33 (0.27)	0.29 (0.13)	0.58 (0.23)	0.36 (0.28)	0.59 (0.27)
	MSLT patients				
	W	1	2	3	R
1p2_32u_128ep	0.80 (0.19)	0.23 (0.20)	0.59 (0.16)	0.23 (0.24)	0.60 (0.25)
1p2_32u_32ep	0.91 (0.12)	0.36 (0.14)	0.70 (0.14)	0.42 (0.30)	0.65 (0.25)
1p2_32u_8ep	0.89 (0.17)	0.29 (0.21)	0.67 (0.15)	0.37 (0.34)	0.65 (0.26)
1p2p1_32u_128ep	0.80 (0.19)	0.23 (0.20)	0.59 (0.16)	0.23 (0.24)	0.60 (0.25)
1p2p1_32u_8ep	0.88 (0.15)	0.30 (0.21)	0.73 (0.17)	0.43 (0.30)	0.59 (0.32)
1p2p1_32u_8ep_res	0.91 (0.11)	0.31 (0.20)	0.55 (0.19)	0.50 (0.29)	0.47 (0.39)
1p_32u_8ep	0.26 (0.24)	0.13 (0.08)	0.29 (0.17)	0.07 (0.18)	0.54 (0.37)

Suppl. Table 3.4. This table represents F1 values of the raw data based algorithms, which were trained on the subjects dataset, i.e. the same models as in the Suppl. Table 3.1, but they were validated on the patient dataset. Sleep recordings (top part) and MSLT recordings (bottom part) were analyzed separately. Standard deviations are shown in bracket. For the meaning of the short names see the corresponding figure.

	Sleep healthy participants				
	W	1	2	3	R
f_128u_128ep	0.80 (0.19)	0.47 (0.21)	0.90 (0.03)	0.85 (0.09)	0.90 (0.10)
f_8u_8ep	0.75 (0.18)	0.31 (0.14)	0.81 (0.06)	0.32 (0.18)	0.89 (0.08)
RF	0.73 (0.14)	0.32 (0.14)	0.88 (0.02)	0.87 (0.08)	0.86 (0.04)
RF_HMM	0.88 (0.07)	0.30 (0.19)	0.92 (0.03)	0.90 (0.06)	0.92 (0.03)
RF_MF	0.83 (0.10)	0.34 (0.16)	0.90 (0.02)	0.88 (0.08)	0.90 (0.03)
	Sleep patients				
	W	1	2	3	R
f_128u_128ep	0.70 (0.08)	0.35 (0.10)	0.72 (0.19)	0.71 (0.35)	0.79 (0.13)
f_8u_8ep	0.59 (0.21)	0.19 (0.15)	0.67 (0.16)	0.26 (0.31)	0.77 (0.19)
RF	0.66 (0.10)	0.24 (0.14)	0.72 (0.21)	0.67 (0.34)	0.65 (0.21)
RF_HMM	0.71 (0.09)	0.16 (0.18)	0.72 (0.21)	0.67 (0.35)	0.72 (0.27)
RF_MF	0.67 (0.09)	0.24 (0.15)	0.73 (0.20)	0.69 (0.35)	0.66 (0.24)
	MSLT patients				
	W	1	2	3	R
f_128u_128ep	0.96 (0.02)	0.43 (0.15)	0.77 (0.09)	0.61 (0.29)	0.80 (0.10)
f_8u_8ep	0.96 (0.04)	0.42 (0.17)	0.69 (0.17)	0.00 (0.00)	0.68 (0.24)
RF	0.95 (0.02)	0.27 (0.10)	0.68 (0.12)	0.31 (0.28)	0.63 (0.13)
RF_HMM	0.97 (0.02)	0.28 (0.14)	0.72 (0.13)	0.18 (0.30)	0.78 (0.10)
RF_MF	0.96 (0.02)	0.28 (0.13)	0.69 (0.12)	0.29 (0.29)	0.70 (0.13)

Suppl. Table 3.5. This table represents F1 values of the feature based algorithms which were trained on the both subjects dataset and patient dataset training parts. Then they were validated on the corresponding test parts. Cross validation and test parts of subjects dataset were merged. The table divided into three parts: sleep recordings of subjects (top part),sleep recordings of patients (middle part) and MSLT recordings (bottom part). Standard deviations are shown in bracket. For the meaning of the short names see the corresponding figure.

	Sleep healthy participants				
	W	1	2	3	R
1p2_32u_128ep	0.61 (0.25)	0.33 (0.17)	0.91 (0.02)	0.92 (0.02)	0.92 (0.03)
1p2_32u_32ep	0.83 (0.07)	0.53 (0.14)	0.91 (0.02)	0.88 (0.05)	0.94 (0.02)
1p2_32u_8ep	0.57 (0.29)	0.31 (0.09)	0.88 (0.03)	0.92 (0.04)	0.90 (0.06)
1p2p1_32u_128ep	0.76 (0.15)	0.46 (0.12)	0.92 (0.01)	0.92 (0.03)	0.95 (0.02)
1p2p1_32u_8ep	0.78 (0.25)	0.44 (0.17)	0.91 (0.02)	0.92 (0.03)	0.93 (0.03)
1p2p1_32u_8ep_res	0.85 (0.18)	0.46 (0.14)	0.90 (0.03)	0.92 (0.03)	0.92 (0.02)
1p_32u_8ep	0.59 (0.31)	0.40 (0.12)	0.89 (0.03)	0.92 (0.03)	0.93 (0.03)
	Sleep patients				
	W	1	2	3	R
1p2_32u_128ep	0.69 (0.13)	0.36 (0.16)	0.77 (0.13)	0.78 (0.17)	0.77 (0.09)
1p2_32u_32ep	0.72 (0.13)	0.42 (0.15)	0.77 (0.14)	0.79 (0.16)	0.78 (0.11)
1p2_32u_8ep	0.57 (0.20)	0.41 (0.13)	0.62 (0.26)	0.80 (0.22)	0.75 (0.14)
1p2p1_32u_128ep	0.65 (0.18)	0.37 (0.17)	0.77 (0.13)	0.78 (0.23)	0.76 (0.11)
1p2p1_32u_8ep	0.60 (0.17)	0.35 (0.16)	0.77 (0.13)	0.81 (0.14)	0.83 (0.08)
1p2p1_32u_8ep_res	0.63 (0.19)	0.36 (0.16)	0.71 (0.19)	0.77 (0.24)	0.80 (0.08)
1p_32u_8ep	0.64 (0.14)	0.37 (0.11)	0.63 (0.27)	0.77 (0.24)	0.57 (0.32)
	MSLT patients				
	W	1	2	3	R
1p2_32u_128ep	0.95 (0.06)	0.38 (0.16)	0.73 (0.14)	0.50 (0.30)	0.73 (0.18)
1p2_32u_32ep	0.97 (0.02)	0.43 (0.17)	0.72 (0.12)	0.57 (0.26)	0.76 (0.10)
1p2_32u_8ep	0.96 (0.02)	0.37 (0.12)	0.51 (0.19)	0.59 (0.31)	0.66 (0.24)
1p2p1_32u_128ep	0.97 (0.02)	0.44 (0.16)	0.74 (0.12)	0.51 (0.26)	0.80 (0.08)
1p2p1_32u_8ep	0.96 (0.03)	0.50 (0.17)	0.77 (0.11)	0.66 (0.26)	0.79 (0.09)
1p2p1_32u_8ep_res	0.94 (0.07)	0.42 (0.19)	0.60 (0.15)	0.52 (0.33)	0.69 (0.12)
1p_32u_8ep	0.97 (0.01)	0.31 (0.19)	0.48 (0.22)	0.65 (0.28)	0.57 (0.32)

Suppl. Table 3.6. This table represents F1 values of the raw data based algorithms which were trained on the both subjects dataset and patient dataset training parts. Then they were validated on the corresponding test parts. Cross validation and test parts of subjects dataset were merged. The table divided into three parts: sleep recordings of subjects (top part), sleep recordings of patients (middle part) and MSLT recordings (bottom part). Standard deviations are shown in bracket. For the meaning of the short names see the corresponding figure.

4 Microsleep episode detection

Jelena Skorucak[1*], David R. Schreier[2*], Anneke Hertig-Godeschalk[2], Alexander Malafeev[1], Johannes Mathis[2#], Peter Achermann[1#]

[1] Institute of Pharmacology and Toxicology, University of Zurich, Switzerland
[2] Sleep-Wake-Epilepsy-Centre, Department of Neurology, Inselspital, Bern University Hospital, and University of Bern, Switzerland

* authors contributed equally
[#] shared last authorship

Acknowledgments and funding
This work was supported by the Swiss National Science Foundation (SNSF, grants 32003B_176323 and 32003B_146643), nano-tera.ch (grant 20NA21_145929), Clinical Research Priority Program “Sleep and Health” of the University of Zurich, and the Swiss Commission of Technology and Innovation (CTI; grant 17864.1 PFLS-LS).

Author contributions: JS, AM and PA designed the analyses; JS conducted the analyses; AM provided consultations on machine learning; DS, AH-G and JM collected and scored the data.

Text below was written by AM and corrected by PA.

4.1 Introduction

I participated in the project dedicated to the detection of microsleep episodes. Microsleep episodes (MSE) are short fragments of sleep lasting 3 to 15 s. Individuals fail to respond to sensory stimuli during MSE. Microsleep episodes often occur in sleep deprived people, and individuals who had insufficient sleep or under boring conditions. MSE are also common in patients with hypersomnia, sleep apnea and narcolepsy due to excessive daytime sleepiness.

The occurrence of microsleep episodes is commonly investigated in the driving simulator. Expert scores MSE visually in the recorded data. MSE are characterized by a change in oscillatory activity in the EEG. Despite the fact that microsleeps are routinely scored in many hospitals there are no established scoring rules for MSE scoring yet. Visual scoring of microsleep is a time demanding process. Thus, development of an automatic tool to perform this task would bring a lot of benefit to clinicians.

4.2 Data and methods

We worked with the data recorded at Sleep-Wake-Epilepsy-Centre of the University Hospital Inselspital in Bern. Seven male and 6 female patients (13 in total; mean age = 33.4 ± 17.1 years) were investigated. Polysomnographic and video recordings of 40 min-long maintenance of wakefulness tests (MWT) were analyzed. The MWT was conducted in a darkened room and patients were instructed not to fall asleep. Microsleep episodes were scored by experts using EEG signal and video recording in order to take eye closure into account.

We extracted 7 features from the EEG and EOG recordings (Table 4.1). Example of the recording and features is presented on the Figure 4.1 (Skorucak, 2017).

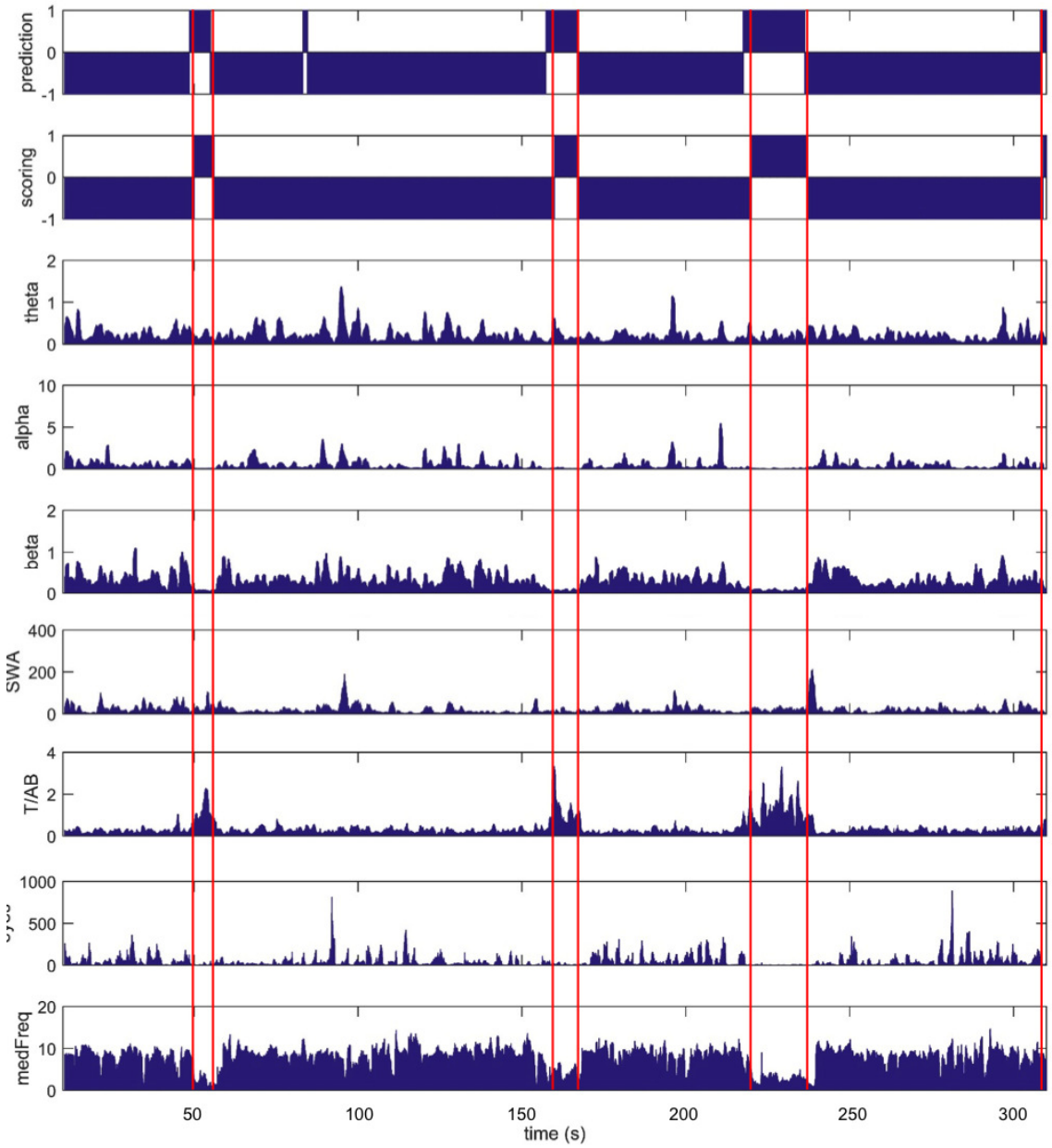


Figure 4.1. Seven features used for the classification of microsleep episodes. Top panel illustrates microsleeps predicted with random forest. Second panel shows microsleeps scored by an expert. MSE are marked with 1 and -1 corresponds to the absence of MSE (Skorucak, 2017). The features are summarized in Table 4.1

Power spectra was computed using an autoregressive model (order 16) on 1-s sliding window with steps of 200 ms. This method allows to capture oscillatory events with a fine time resolution, but since it is parametric approach it can capture a maximum p/2 peaks in the spectra (p - order of autoregressive model).

We employed two classifiers: random forest (RF, 100 trees) and support vector machine (SVM, radial basis function kernel) to classify these 7-dimentional vectors into two groups: microsleep episodes or absence microsleep. The performance of the algorithms was evaluated with specificity and sensitivity.

Feature name (short name in Fig. 4.1)	Frequency range
Theta power (T)	4–8 Hz
Alpha power (A)	8–12 Hz
Beta power (B)	12–26Hz
Slow Waves Activity (SWA)	0.75–4 Hz
$\frac{Theta}{Alpha+Beta}$ (T/[A+B])	
Eye movements (eyes)	SWA in EOG channel divided by SWA in O2A1 channel
Median EEG frequency (medFreq)	0.75–26 Hz

Table 4.1. Features derived to identify micro sleep episodes. They are based on spectral information of the EEG and EOG. Most features represent power in specific bands and their ratios, except for the median frequency.

4.3 Results

Training was performed on the data of 12 patients. Test set contained data recorded in one patient. Microsleep detection on the test data is illustrated in Figure 4.2 (provided by Jelena Skorucak). We computed specificity and sensitivity of SVM and RF classifiers using expert scoring as

ground truth. We obtained specificity equal to 0.99 for both methods, SVM had sensitivity equal to 0.74 and RF scored 0.72.

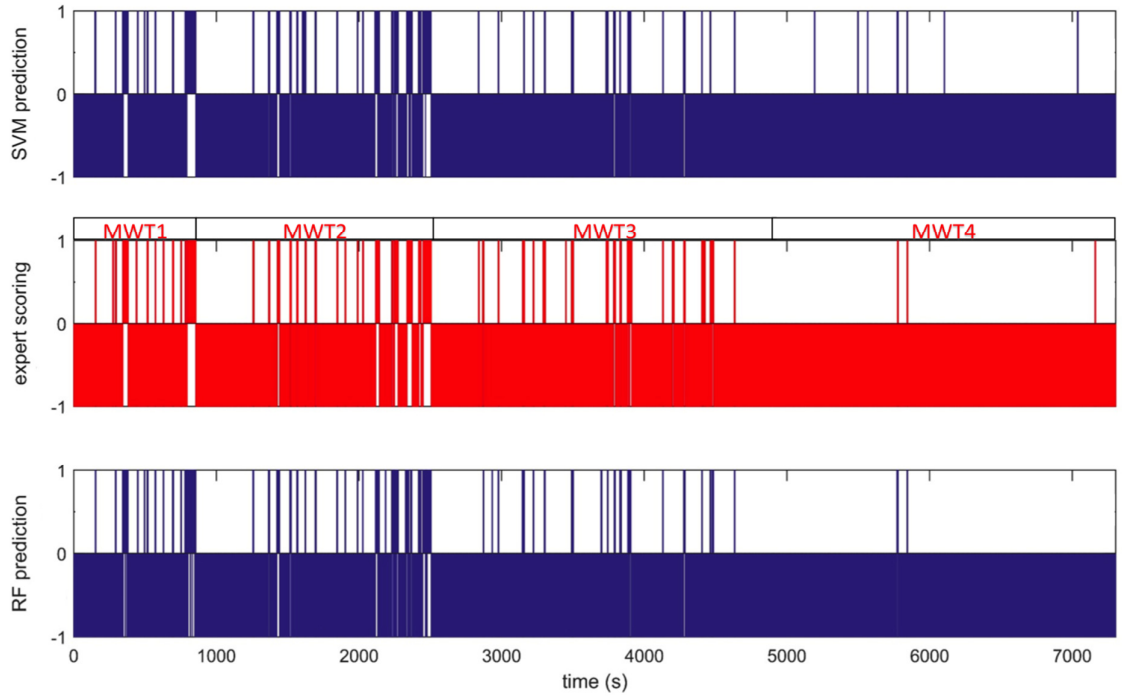


Figure 4.2. Microsleep episodes (MSE) scored by an expert (middle panel, red). 1 corresponds to scored MSE, -1 to the absence of MSE. MWT 1-4 indicates the four recordings of the patient in the test set. Top and bottom panels (blue) show MSE detected by SVM and RF (Skorucak, 2017)

4.4 Conclusion and discussion

We were able to get high specificity and moderately good sensitivity with both SVM and RF methods. However, sensitivity and specificity are not the most optimal metrics to assess the performance of the microsleep detection algorithm. The reason for that is following: a human scorer marks a microsleep episode visually without being able to clearly determine the edges of an episode. Therefore, we naturally will have some discrepancy even if we correctly detect the presence of a microsleep episode.

The main limitation of our study was that we only used one subject to test the models. The methods shall be tested on the larger dataset. If the performance would not be satisfactory one should train the classifiers using a larger dataset. Other approaches might perform better and shall be considered. For example deep learning.

It is possible to include automatic analysis of the video of the subject's face to increase performance. The video is routinely recorded, and the experts use the video to assess the closure of the eyelids.

5 Discussion

Two main topics were addressed in this thesis. The first one is *automatic artifact detection in sleep EEG data.* This is a highly important matter in quantitative EEG analyses since it can be applied to large data sets. This, in its turn, is a key point when it comes to addressing issues and related to genetics, epidemiology, or precision medicine. The second major topic covered in the thesis was *automatic classification of sleep stages*. Recent advantages in machine learning allow the development of reliable classifiers. In the course of this research, multiple algorithms have been developed and were tested on recordings of healthy participants and patients.

5.1 Automatic artifact detection

Rejection of epochs contaminated with artifacts is crucial for quantitative sleep EEG analysis. It is impossible to obtain e.g. clean average power density spectra of the sleep EEG without the artifact exclusion. We systematically evaluated and compared sets of simple algorithms, which were applied in the literature. Our results revealed that such methods perform well. They had moderate to good sensitivity (TPR). Specificity (1 - FPR) was fixed at 0.9 by the design in order to choose the thresholds. With most of the methods we tested, we were able to get clean average power density spectra. Automatic artifact rejection is a very useful data processing step since manual artifact exclusion is time consuming (Anderer et al., 1999, Coppieters't Wallant et al., 2016).

Next to data of healthy participants, we applied our algorithms to sleep and MSLT data recorded in narcoleptic and hypersomniac patients. The outcome of artifact detection by our methods on the new dataset was good

which suggests that the algorithms we evaluated had a good generalization capacity.

We noticed that methods detecting high power in upper frequency range work better than others. It can be explained by the fact that most muscle artifacts are characterized by the power in the high frequency range (Goncharova et al., 2003).

The simplicity of our algorithms have a downside though. These algorithms are not flexible enough to produce both high sensitivity and specificity. Since we fixed specificity at 0.9, our methods excluded on average 16.3% of epochs whereas the experts excluded only 7% of them. This problem can be solved by developing a more sophisticated algorithm with both high specificity and sensitivity.

As long as our methods reacted to the high frequency power of the signal, we were unable to differentiate between artifacts contaminating the whole spectra and those contaminating only high frequency range. For sleep research it would be quite useful not to exclude epochs which contain an artifact in the high frequency range but, at the same time, do not affect the area of interest (0.5-20 Hz). This could be achieved by using more than one parameter to characterize an epoch. In this case dimensionality of the data will increase. This way we will need a machine learning algorithm in order to solve such a problem.

The downside of more complex models is that they can learn properties of a particular dataset (e.g. healthy subjects). Classification quality might suffer in case another dataset (e.g. patients) does not have these properties. This phenomenon is called overfitting. Therefore, it is necessary to find a good compromise between the complexity of the model and prediction accuracy (Geman et al., 1992).

Simple threshold-based methods might produce bad results if they are applied to the EEG of children, infants, or people under the influence of medication. The reason for this is the altered EEG amplitude under these conditions. For example, children have much larger slow waves. We expect that machine learning-based methods would handle such conditions better because they can take more information into account.

The artifact rejection problem could also be addressed with machine learning. I would start from using the spectra as a feature vector and classify it as either an artifact or a clean epoch. Any of the good classification algorithms could in principle be used. I would prefer the random forest (RF) classifier. We already have preliminary results and the quality of classification was very high. The reason is that the RF has a good performance in many applications and its robustness to the noisy data (Breiman et al., 1984). In order to detect artifacts, we could also employ convolutional neural networks.

I think the key for high quality of automatic artifact detection is a good dataset. It should include broad variety of subjects, as well as being scored by several (preferably 3) experts. This way we can reach a consensus among the experts and avoid human errors.

We focused on single EEG derivations because we intended to work with systems recording only a small number of channels (datasets used in this thesis had 12 and 6 EEG channels). Our algorithms were meant to be used with data recorded with a portable EEG device with four channels.

In general, if possible, it might be better to work with high density EEG data. In such a setting, artifacts can be subtracted instead of rejecting contaminated epochs (Delorme et al., 2007). Of course, this can distort the data. It has to be ensured that data distortion does not affect the main purpose of the analyses. ICA requires generally visual identification of

artifactual components. It could be done using machine learning as well. Because a human classifies them visually, I expect that classification with CNN would work well on this task. However, the collection of the training dataset will be the biggest challenge in creating such a classifier.

5.2 Sleep stage classification

The largest part of this work was dedicated to automatic sleep scoring. We were able to achieve good performance of the automatic scoring algorithm using several machine learning methods, namely, random forest and multiple artificial neural networks. We were able to perform automatic scoring with both carefully engineered features and raw data. Methods working with raw data showed a slightly better performance than the feature-based ones.

We used F1 scores to assess the performance of our models. The F1 scores we reached with our methods were comparable to the in agreement rate of experts (Danker-Hopfe et al., 2004, Penzel et al., 2013, Rosenberg and Van Hout, 2013).

I think F1 score is not the best method to assess the quality of sleep scoring. Let us assume that an algorithm made an episode of a certain stage either several epochs longer or shorter. It is unlikely to affect any clinical assessment. On the contrary, if an algorithm (or a human expert) misses a REM sleep episode at the sleep onset (SOREM sleep episode), it might have an impact on the diagnosis. Thus, a score that takes the temporal sequence of stages into account would be needed.

In my opinion, quality of our automatic scoring is not yet sufficient enough to be applied in a clinical setting. I expect that the performance of our algorithms on the data of patients with different disorders (not present in the

training dataset) might be worse than in our study. Automatic scoring requires further development and testing. I believe that automatic scoring models can greatly benefit from a larger amount of training data and examples scored by several experts. Large amounts of training data, especially, when the data are collected in diverse types of patients and laboratories, creates an opportunity for the algorithms to learn how to deal with the most of possible cases.

Labels produced by independent scorers are needed to avoid ambiguous information in the training data. It is well known that experts do not have a perfect agreement between each other (Danker-Hopfe et al., 2004, Penzel et al., 2013, Rosenberg and Van Hout, 2013, Younes et al., 2016, Younes et al., 2018). We could then take consensus or average labels for stages. In this way, 'difficult' epochs will have a lower weight.

The biggest challenge for automatic scoring algorithms is the same as for human experts: sleep of the subjects with sleep issues, medication altered sleep and noisy data. I expect that deep learning methods may learn patterns of altered sleep. Of course, examples of such sleep shall be present in the training set and I would expect better results when networks are trained on very large datasets, including diverse patient data.

Feature-based methods have certain advantages: they are fast and able to be trained on small datasets. Feature engineering however, is a very time-consuming task. On the other hand, features may help to understand underlying mechanisms and algorithms might give information on the most important features for classification, whereas artificial neural networks do not necessarily provide insights into their classification strategy.

For our sleep classification algorithm, we also developed a complex wavelet-based algorithm for eye movement detection. It required tremendous efforts. We used thresholds which were manually set, based on our experience

and common sense. This eye movement detection algorithm would benefit from threshold optimization. To do so, it would require a dataset where the different eye movements are marked by an expert. On the other hand, deep learning approaches do not require any feature engineering. Thus, using deep learning methods, we do not need to have a dataset with marked eye movements. Deep neural network would learn that eye movements are relevant on their own.

Yet another major advantage of deep learning-based methods is the simplicity of the code which is easier to maintain than those complex feature extraction modules. Deep learning, however, requires more computational resources for training than the feature-based ones.

In our study we observed that EOG and EMG signals are helpful for automatic sleep scoring. This is not surprising at all since they carry important information about eye movements and muscle tone and are part of the scoring rules (Rechtschaffen and Kales, 1968, Iber et al., 2007). If these signals are noisy, though, the algorithms gets confused resulting in an overall performance deterioration. Similar observations were made by SIESTA group (Anderer et al., 2005). Taking all the information into consideration, I think an automatic scoring systems would benefit from an automatic data quality checking. It is also possible to detect noisy channels using machine learning. However, visual review of the data by an expert will remain essential.

I would suggest using all available channels as input when they are clean and use models with reduced number of input channels when some channels are noisy. We observed that CNN-LSTM networks produced good results even using a single EEG channel.

As it has been previously mentioned in our study, we found that deep neural net performs very well even with a single EEG channel. It was also

observed in the literature (Tsinalis et al., 2016). It is surprising since reliably distinguishing between REM sleep and quiet wakefulness using only a single EEG channel is difficult for a human expert too. To score REM sleep, humans mostly rely on rapid eye movements and low muscle tone (Rechtschaffen and Kales, 1968). I assume that to identify REM sleep neural networks use such patterns as saw-tooth waves which were found to be one of the markers of REM sleep in animals (Jouvet et al., 1960) and humans (Takahara et al., 2009).

I would also include some prior information into the scoring algorithm. Sleep scoring rules are to some degree subjective, but they do have very clear criteria for the scoring of deep sleep: slow waves are supposed to cover certain time-period within an epoch (more than 20% of an epoch according to the AASM rules) (Iber et al., 2007). And the criteria for slow wave detection are also well-known (amplitudes larger than 75 µVolt peak-to-peak and a duration longer than 0.5 s) (Iber et al., 2007). Unfortunately, these criteria are not always followed by a human expert (Younes et al., 2018). It would mean that we classify NREM sleep (stage 2 and 3) only and then make the dissociation based on slow wave criteria. Scoring deep sleep based on threshold for the slow wave amount has been implemented previously in an automatic scoring system (Malhotra et al., 2013).

When it comes to automatic scoring, one of the biggest challenges is the presence of movement and muscle artifacts in the recording. In the scoring rules according to Rechtschaffen and Kales (Rechtschaffen and Kales, 1968) such events belong to a separate class called movement time (MT). This makes sense because an expert can not see anything useful in presence of a strong movement or muscle artifacts. In the newer AASM scoring rules (Iber et al., 2007), MT was abolished, and a movement must be associated to the most plausible sleep stage. As a consequence, such stages are contaminated by

artifacts which makes automatic scoring difficult if not impossible. It damages the learning process of an algorithm because an algorithm receives ambiguous information regarding such epochs. To my view, scoring epochs with large artifacts as a separate stage would be very beneficial.

I think it would be very interesting to train a LSTM network on raw data for sleep classification without the convolutional part. One of the challenges is that we have only one sleep stage label per scoring epoch, i.e. for all samples within the epoch. This problem could be avoided if we would e.g. use only the last sample in the epoch to compute the loss function. Let us assume the epoch length is 20 s and sampling rate is 100 Hz. Then, samples 1-199 will not contribute to the loss function and only the 200th sample would. The next sample contributing to the loss function would be the 400th, 600th, and so on. This makes sense because the LSTM network would have had analyzed the whole epoch at the last sample of the epoch, therefore it has all the necessary information to classify it.

But the most interesting thing would be that we could get a classification of the sleep stage not only for the epoch, but for every sample. It might provide novel information on the transitions between stages.

Furthermore, I think that automatic scoring could benefit of high-density EEG data. High-density EEG allows artifact removal using ICA (Delorme et al., 2007) and source reconstruction of brain activity (electromagnetic tomography) (Pascual-Marqui et al., 1994). First of all, data without artifacts are likely to carry more information. Second reconstructed sources using the signals from the whole cortex may carry more information about the state of the brain.

There have been attempts to score sleep using fMRI signal (Tagliazucchi et al., 2012). It suggests that information about activation of brain regions is

sufficient to score sleep. I expect that scoring using electromagnetic tomography and deep learning can be better than scoring using just signals from several channels. However, the computational effort of source localization and artifact removal for the entire night would be very high.

As our preliminary analysis revealed, machine learning also works for microsleep episode detection. I expect that LSTM networks would bring a further improvement since such networks are able to take local temporal information into account. Working with raw data could also be advantageous in case our features are not optimal which does not seem to be the case. Surprisingly, we could reliably detect microsleep episodes using only 7 features.

The main disadvantage of our approach was that we had very limited dataset. We trained on the data of 12 patients and tested the algorithms on the data of a single patient. The dataset shall be extended and preferably scored by several independent experts.

Expert scores edges of the microsleep with certain precision. This might create ambiguous information for the machine learning algorithms. It might be helpful to reduce the importance of the data points close to the edges of microsleep episodes, i.e. give them a lower weight for training.

Our detection of microsleep episodes was mainly based on the EEG. However, a video of the face of the subject is also routinely recorded and an expert uses this information to estimate the closure of the eyes. It would also possible to incorporate video recordings into the machine learning algorithm in case the performance of the algorithms using PSG signals might not be sufficient.

Attempts to score sleep in an unsupervised manner have been undertaken in both humans and animals (Agarwal and Gotman, 2001, Grube et al., 2002, Gath and Geva, 1989, Sunagawa et al., 2013, Libourel et al., 2015).

Recently unsupervised learning gained a lot of attention, especially in the area of image processing (Le, 2013, Oord et al., 2016) and time-series analyses, namely natural language processing (NLP) (Mikolov et al., 2010, Conneau et al., 2017, Salakhutdinov and Hinton, 2009, Artetxe et al., 2017, Lample et al., 2017).

We have already used unsupervised learning, specifically for clustering in our artifact detection work. It did not work well with a low proportion of artifacts, but there are novel more sophisticated approaches to solve unsupervised learning problem. In particular, I would consider the autoencoder (Hinton and Salakhutdinov, 2006) or the variational autoencoder (VAE) (Kingma and Welling, 2013).

Machine translation using unsupervised learning was especially successful because machine translation suffers from the lack of parallel texts (same text in different languages), especially for languages with a low number of speakers. Generally, it is better to use supervised learning when enough of well labeled data are available. Otherwise one may use unsupervised learning.

In case of sleep it might be a good idea because the scoring provided by an expert is often subjective and there is limited agreement between scorers (Danker-Hopfe et al., 2004, Penzel et al., 2013, Rosenberg and Van Hout, 2013, Younes et al., 2016, Younes et al., 2018). Moreover, with unsupervised learning we could get rid of scoring epochs and score sleep continuously. We might also revise the sleep stages, it might be that we would discover new features by looking at the internal representation of the autoencoder. Another interesting

application are generative models. Variational autoencoder (VAE) (Kingma and Welling, 2013) allows the generation of a signal that could be used for example to generate sleep data for educational reasons.

Unsupervised learning is likely to be a good idea for microsleep episode detection because it is very difficult to collect and score these data. Available datasets are small, and again we could avoid the problem of subjective scoring. Moreover, as was already mentioned a human expert defines the borders of a microsleep episode with limited precision. This problem is not relevant for unsupervised learning.

It would be also possible to approach microsleep episodes detection with semi-supervised learning. Microsleeps are rare events in a recording. One could train an autoencoder on the data without microsleeps in order to learn features of the data and then transfer the weights of the encoder part into a new network which solves a classification problem (Erhan et al., 2010): classification as a microsleep episode or wakefulness. This network would be already pretrained, and we just would need to fine-tune it using small amounts of labeled data.

I think that the problem of artifact detection could also be addressed with unsupervised learning. The artifact in the sleep data could be considered as an anomaly. This idea has already been used for artifact detection with an autoregressive model (AR) (Schlögl, 2000). The idea is to predict the signal using previous samples and look at the discrepancy between predicted and measured signal. If the discrepancy is big it means that an anomaly (artifact) is present, i.e. we say that we have an artifact in the signal if the real signal deviates from the model prediction of a physiological signal.

This approach is common for anomaly detection Chandola et al. (2009), and has already been applied to ECG data (alDosari, 2016). They used a LSTM

autoencoder in order to model the signal. We expect that LSTM autoencoder would perform well for the artifact detection in the EEG signal as well.

Autoencoders naturally reduce the dimensionality of the data. An autoencoder could be also trained to reproduce the signal from a signal artificially contaminated with noise (Vincent et al., 2008). It is called Denoising Autoencoder (DAE). It might be even possible to remove certain artifacts from the EEG signal with the use of a denoising autoencoders. However, it raises the same problem as artifact removal with ICA, we do not know the “true” signal and the resulting signal might be distorted.

In this work I have shown that modern machine learning methods, especially, deep neural networks, are useful in the field of sleep research. In my opinion the great advantage of deep learning is the fact that one can omit the feature-engineering step. Feature engineering is the most time consuming part. Deep neural networks can be programmed quickly and applied to a broad variety of classification problems. The biggest challenge is the same as for classical machine learning methods – data collection and segmentation by an expert.

Bibliography

Abadi, M., Barham, P., Chen, J. *et al.* TensorFlow: A System for Large-Scale Machine Learning. *OSDI*, 2016: 265-283.

Abdel-Hamid, O., Mohamed, A.-R., Jiang, H., Deng, L., Penn, G. and Yu, D. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 2014, 22: 1533-1545.

Achermann, P. Sleep. In: M. AKAY (Ed.), *Wiley Encyclopedia of Biomedical Engineering*. John Wiley & Sons, Inc., 2006.

Achermann, P., Malafeev, A., Skorucak, J. and Tarokh, L. Automated Sleep Analysis. *IEEE EMBC*, Milano, 2015.

Achermann, P. and Tarokh, L. Human sleep and its regulation. *Kosmos*, 2014, 2: 173-180.

Aeschbach, D. and Borbély, A. All-night dynamics of the human sleep EEG. *Journal of sleep research*, 1993, 2: 70-81.

Agarwal, R. and Gotman, J. Computer-assisted sleep staging. *IEEE Transactions on Biomedical Engineering*, 2001, 48: 1412-1423.

Al-Rfou, R., Alain, G., Almahairi, A. *et al.* Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint*, 2016.

Aldosari, M. S. Unsupervised Anomaly Detection in Sequences Using Long Short Term Memory Recurrent Neural Networks. In, 2016.

Alhola, P. and Polo-Kantola, P. Sleep deprivation: Impact on cognitive performance. *Neuropsychiatric disease and treatment*, 2007, 3: 553.

Altman, D. G. and Bland, J. M. Diagnostic tests. 1: Sensitivity and specificity. *BMJ: British Medical Journal*, 1994, 308: 1552.

Anderer, P., Gruber, G., Parapatics, S. *et al.* An E-health solution for automatic sleep classification according to Rechtschaffen and Kales: validation study of the Somnolyzer 24× 7 utilizing the Siesta database. *Neuropsychobiology*, 2005, 51: 115-133.

Anderer, P., Roberts, S., Schlögl, A. *et al.* Artifact processing in computerized analysis of sleep EEG - A review. *Neuropsychobiology*, 1999, 40: 150-157.

Andrew, N. Sparse autoencoder. *CS294A Lecture notes*, 2011, 72.

Arlot, S. and Celisse, A. A survey of cross-validation procedures for model selection. *Statistics surveys*, 2010, 4: 40-79.

Artetxe, M., Labaka, G., Agirre, E. and Cho, K. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*, 2017.

Aserinsky, E. and Kleitman, N. Regularly occurring periods of eye motility, and concomitant phenomena, during sleep. *Science*, 1953, 118: 273-274.

Banks, S. and Dinges, D. F. Behavioral and physiological consequences of sleep restriction. *Journal of clinical sleep medicine: JCSM: official publication of the American Academy of Sleep Medicine*, 2007, 3: 519.

Barlow, J. S. Muscle spike artifact minimization in EEGs by time-domain filtering. *Electroencephalography and Clinical Neurophysiology*, 1983, 55: 487-491.

Barlow, J. S. EMG artifact minimization during clinical EEG recordings by special analog filtering. *Electroencephalography and Clinical Neurophysiology*, 1984, 58: 161-174.

Barlow, J. S. Automatic Elimination of Electrode-Pop Artifacts. *IEEE Transactions on Biomedical Engineering*, 1986, BME-33: 517-521.

Bengio, Y., Boulanger-Lewandowski, N. and Pascanu, R. Advances in optimizing recurrent networks. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013: 8624-8628.

Berger, H. Über das elektrenkephalogramm des menschen. *European Archives of Psychiatry and Clinical Neuroscience*, 1929, 87: 527-570.

Bersagliere, A. and Achermann, P. Slow oscillations in human non-rapid eye movement sleep electroencephalogram: effects of increased sleep pressure. *Journal of sleep research*, 2010, 19: 228-237.

Bersagliere, A., Pascual-Marqui, R. D., Tarokh, L. and Achermann, P. Mapping slow waves by EEG topography and source localization: Effects of sleep deprivation. *Brain topography*, 2018, 31: 257-269.

Bishop, C. *Pattern recognition and machine learning*. Springer-Verlag New York, 2016.

Bixler, E. O., Kales, A., Soldatos, C. R., Kales, J. D. and Healey, S. Prevalence of sleep disorders in the Los Angeles metropolitan area. *The American Journal of Psychiatry*, 1979.

Bodenstein, G. and Praetorius, H. M. Feature extraction from the electroencephalogram by adaptive segmentation. *Proceedings of the IEEE*, 1977, 65: 642-52.

Borbély, A. and Achermann, P. Sleep homeostasis and models of sleep regulation. *Journal of biological rhythms*, 1999, 14: 559-570.

Borbély, A., Tobler, I., Achermann, P. and Geering, B. Bits of Sleep CD-ROM. Sleep Research Laboratory of the University of Zurich, Institute of Pharmacology, Sleep Research Laboratory of the University of Zurich, Institute of Pharmacology, 1998.

Borbély, A. A., Baumann, F., Brandeis, D., Strauch, I. and Lehmann, D. Sleep deprivation: effect on sleep stages and EEG power density in man. *Electroencephalography and clinical neurophysiology*, 1981, 51: 483-493.

Born, J., Rasch, B. and Gais, S. Sleep to remember. *The Neuroscientist*, 2006, 12: 410-424.

Breiman, L. Random forests. *Machine learning*, 2001, 45: 5-32.

Breiman, L., Friedman, J., Stone, C. J. and Olshen, R. A. *Classification and regression trees*. CRC press, 1984.

Brunner, D. P., Dijk, D.-J., Münch, M. and Borbély, A. A. Effect of zolpidem on sleep and sleep EEG spectra in healthy young men. *Psychopharmacology*, 1991, 104: 1-5.

Buckelmüller, J., Landolt, H.-P., Stassen, H. and Achermann, P. Trait-like individual differences in the human sleep electroencephalogram. *Neuroscience*, 2006, 138: 351-356.

Buzsáki, G., Anastassiou, C. A. and Koch, C. The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes. *Nature reviews neuroscience*, 2012, 13: 407-420.

Cauchy, A. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 1847, 25: 536-538.

Cecotti, H. and Graeser, A. Convolutional neural network with embedded Fourier transform for EEG classification. *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008: 1-4.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016: 785-794.

Chollet, F. a. O. Keras. GitHub, 2015.

Cirelli, C. and Tononi, G. Is sleep essential? *PLoS biology*, 2008, 6: e216.

Clugston, G. and Garlick, P. The response of whole-body protein turnover to feeding in obese subjects given a protein-free, low-energy diet for three weeks. *Human nutrition. Clinical nutrition*, 1982, 36: 391-397.

Cluitmans, P. M., Jansen, J. and Beneken, J. W. Artifact detection and removal during auditory evoked potential monitoring. *Journal of Clinical Monitoring*, 1993, 9: 112-120.

Comon, P. Independent component analysis, A new concept? *Signal Processing*, 1994, 36: 287-314.

Conneau, A., Lample, G., Ranzato, M. A., Denoyer, L. and Jégou, H. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

Copinschi, G., Leproult, R. and Spiegel, K. The important role of sleep in metabolism. *How Gut and Brain Control Metabolism*. Karger Publishers, 2014: 59-72.

Coppieters't Wallant, D., Muto, V., Gaggioni, G. *et al.* Automatic artifacts and arousals detection in whole-night sleep EEG recordings. *Journal of Neuroscience Methods*, 2016, 258: 124-133.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 1995, 20: 273-297.

Cox, D. R. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1958: 215-242.

D'rozario, A. L., Dungan Ii, G. C., Banks, S. *et al.* An automated algorithm to identify and reject artefacts for quantitative EEG analysis during sleep in patients with sleep-disordered breathing. *Sleep and Breathing*, 2015, 19: 607-615.

Danker-Hopfe, H., Anderer, P., Zeitlhofer, J. *et al.* Interrater reliability for sleep scoring according to the Rechtschaffen & Kales and the new AASM standard. *Journal of sleep research*, 2009, 18: 74-84.

Danker-Hopfe, H., Kunz, D., Gruber, G. *et al.*Interrater reliability between scorers from eight European sleep laboratories in subjects with different sleep disorders. *Journal of sleep research*, 2004, 13: 63-69.

Davidson, P., Jones, R. and Peiris, M. Detecting behavioral microsleeps using EEG and LSTM recurrent neural networks. *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*. IEEE, 2006: 5754-5757.

De Boer, P.-T., Kroese, D. P., Mannor, S. and Rubinstein, R. Y. A tutorial on the cross-entropy method. *Annals of operations research*, 2005, 134: 19-67.

De Gennaro, L., Marzano, C., Fratello, F. *et al.* The electroencephalographic fingerprint of sleep is genetically determined: a twin study. *Annals of neurology*, 2008, 64: 455-460.

Delorme, A. and Makeig, S. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of neuroscience methods*, 2004, 134: 9-21.

Delorme, A., Sejnowski, T. and Makeig, S. Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis. *NeuroImage*, 2007, 34: 1443-1449.

Dice, L. R. Measures of the amount of ecologic association between species. *Ecology*, 1945, 26: 297-302.

Diekelmann, S. and Born, J. The memory function of sleep. *Nature Reviews Neuroscience*, 2010, 11: 114.

Doroshenkov, L., Konyshev, V. and Selishchev, S. Classification of human sleep stages based on EEG processing using hidden Markov models. *Biomedical Engineering*, 2007, 41: 25-28.

Dos Santos, C. and Gatti, M. Deep convolutional neural networks for sentiment analysis of short texts. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014: 69-78.

Drummond, J., Brann, C., Perkins, D. and Wolfe, D. A comparison of median frequency, spectral edge frequency, a frequency band power ratio, total power, and dominance shift in the determination of depth of anesthesia. *Acta Anaesthesiologica Scandinavica*, 1991, 35: 693-699.

Du Bois-Reymond, E. *Untersuchungen über thierische Elektricität*. G. Reimer, 1848.

Durka, P.J., Klekowicz, H., Blinowska, K. J., Szelenberger, W. and Niemcewicz, S. A simple system for detection of EEG artifacts in polysomnographic recordings. *IEEE Transactions on Biomedical Engineering*, 2003, 50: 526-528.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P. and Bengio, S. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 2010, 11: 625-660.

Everett, B. *An introduction to latent variable models*. Springer Science & Business Media, 2013.

Everson, C. A., Bergmann, B. M. and Rechtschaffen, A. Sleep deprivation in the rat: III. Total sleep deprivation. *Sleep*, 1989, 12: 13-21.

Faraut, B., Bayon, V. and Léger, D. Neuroendocrine, immune and oxidative stress in shift workers. *Sleep medicine reviews*, 2013, 17: 433-444.

Farley, B. and Clark, W. Simulation of self-organizing systems by digital computer. *Transactions of the IRE Professional Group on Information Theory*, 1954, 4: 76-84.

Fattinger, S., De Beukelaar, T. T., Ruddy, K. L. *et al.* Deep sleep maintains learning efficiency of the human brain. *Nature Communications*, 2017, 8: 15405.

Fell, J., Röschke, J., Mann, K. and Schäffner, C. Discrimination of sleep stages: a comparison between spectral and nonlinear EEG measures. *Electroencephalography and clinical Neurophysiology*, 1996, 98: 401-410.

Flanigan Jr, W., Knight, C., Hartse, K. and Rechtschaffen, A. Sleep and wakefulness in chelonian reptiles. I. The box turtle, Terrapene carolina. *Archives italiennes de biologie*, 1974, 112: 227.

Force, A. O. S. a. T. and Medicine, A. a. O. S. Clinical guideline for the evaluation, management and long-term care of obstructive sleep apnea in adults. *Journal of clinical sleep medicine: JCSM: official publication of the American Academy of Sleep Medicine*, 2009, 5: 263.

Fukushima, K. and Miyake, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. *Competition and cooperation in neural nets*. Springer, 1982: 267-285.

Gaillard, J. and Tissot, R. Principles of automatic analysis of sleep records with a hybrid system. *Computers and biomedical research*, 1973, 6: 1-13.

Gath, I. and Geva, A. B. Unsupervised optimal fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, 1989, 11: 773-780.

Gavelin, R., Klomp, H., Priddle, C. and Uddenfeldt, M. Blind Source Separation–Report for Adaptive Signal Processing Project. *Department of Engineering Sciences, Uppsala University, Sweden*, 2004.

Geman, S., Bienenstock, E. and Doursat, R. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 1992, 4: 1-58.

Gevins, A. S. and Rémond, A. *Methods of analysis of brain electrical and magnetic signals*. Elsevier Science Limited, 1987.

Gevins, A. S., Yeager, C. L., Diamond, S. L., Spire, J., Zeitlin, G. M. and Gevins, A. H. Automated analysis of the electrical activity of the human brain (EEG): A progress report. *Proceedings of the IEEE*, 1975, 63: 1382-1399.

Giedke, H. and Schwärzler, F. Therapeutic use of sleep deprivation in depression. *Sleep medicine reviews*, 2002, 6: 361-377.

Girolami, M. An Alternative Perspective on Adaptive Independent Component Analysis Algorithms. *Neural Computation*, 1998, 10: 2103-2114.

Glorot, X., Bordes, A. and Bengio, Y. Deep sparse rectifier neural networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011: 315-323.

Goncharova, I. I., Mcfarland, D. J., Vaughan, T. M. and Wolpaw, J. R. EMG contamination of EEG: spectral and topographical characteristics. *Clinical Neurophysiology*, 2003, 114: 1580-1593.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y. *Deep learning*. MIT press Cambridge, 2016.

Gotman, J., Ives, J. R. and Gloor, P. Frequency content of EEG and EMG at seizure onset: Possibility of removal of EMG artefact by digital filtering. *Electroencephalography and Clinical Neurophysiology*, 1981, 52: 626-639.

Graves, A., Mohamed, A.-R. and Hinton, G. Speech recognition with deep recurrent neural networks. *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on*. IEEE, 2013: 6645-6649.

Green, D. M. and Swets, J. A. *Signal Detection Theory and Psychophysics*. John Wiley and sons, New York, 1966.

Groppe, D. M., Makeig, S. and Kutas, M. Identifying reliable independent components via split-half comparisons. *NeuroImage*, 2009, 45: 1199-1211.

Grube, G., Flexer, A. and Dorffner, G. Unsupervised continuous sleep analysis. *Methods Find Exp Clin Pharmacol*, 2002, 24: 51-56.

Habibzadeh, F., Habibzadeh, P. and Yadollahie, M. On determining the most appropriate test cut-off value: the case of tests with continuous results. *Biochemia medica*, 2016, 26: 297-307.

He, K., Zhang, X., Ren, S. and Sun, J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016: 770-778.

Heller, H. C. and Ruby, N. F. Sleep and circadian rhythms in mammalian torpor. *Annu. Rev. Physiol.*, 2004, 66: 275-89.

Hendricks, J. C., Finn, S. M., Panckeri, K. A. *et al.* Rest in Drosophila is a sleep-like state. *Neuron*, 2000, 25: 129-138.

Hersberger, K., Renggli, V., Nirkko, A. C., Mathis, J., Schwegler, K. and Bloch, K. Screening for sleep disorders in community pharmacies–evaluation of a campaign in Switzerland. *Journal of clinical pharmacy and therapeutics*, 2006, 31: 35-41.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 2006, 313: 504-507.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Ho, T. K. Random decision forests. *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*. IEEE, 1995: 278-282.

Hobson, J. A. Electrographic correlates of behavior in the frog with special reference to sleep. *Electroencephalography and clinical neurophysiology*, 1967, 22: 113-121.

Hochreiter, S. Untersuchungen zu dynamischen neuronalen Netzen. *Diploma, Technische Universität München*, 1991, 91.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 1997, 9: 1735-1780.

Hubel, D. H. and Wiesel, T. N. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 1959, 148: 574-591.

Hunt, E. B., Marin, J. and Stone, P. J. Experiments in induction. 1966.

Iber, C., Ancoli-Israel, S., Chesson, A. and Quan, S. F. *The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications*. American Academy of Sleep Medicine Westchester, IL, 2007.

Iliff, J. J., Wang, M., Liao, Y. *et al.* A paravascular pathway facilitates CSF flow through the brain parenchyma and the clearance of interstitial solutes, including amyloid β. *Science translational medicine*, 2012, 4: 147ra11-147ra11.

Imtiaz, S. A. and Rodriguez-Villegas, E. A low computational cost algorithm for REM sleep detection using single channel EEG. *Annals of biomedical engineering*, 2014, 42: 2344-2359.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Irwin, M. R. Why sleep is important for health: a psychoneuroimmunology perspective. *Annual review of psychology*, 2015, 66.

Itil, T., Shapiro, D., Fink, M. and Kassebaum, D. Digital computer classifications of EEG sleep stages. *Electroencephalography and clinical neurophysiology*, 1969, 27: 76-83.

Ivakhnenko, A. G. and Lapa, V. G. Cybernetics and forecasting techniques. 1967

Jasper, H. The 10/20 intemational electrode system. *EEG Clinical Neurophysiology*, 1958, 10: 371-375.

Ji, D. and Wilson, M. A. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience*, 2007, 10: 100.

Jouvet, M., Michel, F. and Courjon, J. Sur un stade d'activité électrique cérébrale rapide au cours du sommeil physiologique. *CR Soc Biol*, 1959, 153: 1024-1028.

Jouvet, M., Michel, F. and Mounier, D. *Analyse électroencéphlographique comparée du sommeil physiologique chez le chat et chez l'homme*. 1960.

Karni, A., Tanne, D., Rubenstein, B. S., Askenasy, J. J. and Sagi, D. Dependence on REM sleep of overnight improvement of a perceptual skill. *SCIENCE-NEW YORK THEN WASHINGTON*, 1994: 679-679.

Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions.*Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015: 3128-3137.

Kerkhof, G. and Van Dongen, H. Effects of sleep deprivation on cognition. *Human sleep and cognition: basic research*, 2010, 185: 105.

Kiefer, J. and Wolfowitz, J. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 1952: 462-466.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kjellberg, A. Sleep deprivation and some aspects of performance: II. Lapses and other attentional effects. *Waking & Sleeping*, 1977.

Klaue, R. Die bioelektrische Tätigkeit der Grosshirnrinde im normalen Schlaf und in der Narkose durch Schlafmittel. JA Barth, 1937.

Klosh, G., Kemp, B., Penzel, T. *et al.* The SIESTA project polygraphic and clinical database. *IEEE Engineering in Medicine and Biology Magazine*, 2001, 20: 51-57.

Ktonas, P. Y., Osorio, P. L. and Everett, R. L. Automated detection of EEG artifacts during sleep: Preprocessing for all-night spectral analysis. *Electroencephalography and Clinical Neurophysiology*, 1979, 46: 382-388.

Lample, G., Denoyer, L. and Ranzato, M. A. Unsupervised Machine Translation Using Monolingual Corpora Only. *arXiv preprint arXiv:1711.00043*, 2017.

Längkvist, M., Karlsson, L. and Loutfi, A. Sleep stage classification using unsupervised feature learning. *Advances in Artificial Neural Systems*, 2012, 2012: 5.

Laptev, D. and Buhmann, J. M. Convolutional decision trees for feature learning and segmentation. *German Conference on Pattern Recognition*. Springer, Cham, 2014: 95-106.

Laptev, D. and Buhmann, J. M. Transformation-invariant convolutional jungles. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015: 3043-51.

Larsen, L. and Walter, D. On automatic methods of sleep staging by EEG spectra. *Electroencephalography and clinical neurophysiology*, 1970, 28: 459-467.

Le, Q. V. Building high-level features using large scale unsupervised learning. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013: 8595-8598.

Lecun, Y., Boser, B., Denker, J. S. *et al.* Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989, 1: 541-551.

Lee, T.-W., Girolami, M. and Sejnowski, T. J. Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural computation*, 1999, 11: 417-441.

Lennox, W. G., Gibbs, E. L. and Gibbs, F. A. The brain-wave pattern, an hereditary trait; evidence from 74 "normal" pairs of twins. *Journal of Heredity*, 1945.

Lessard, C. and Paschall, R. A system for quantifying EEG slow wave activity. *Electroencephalography and clinical neurophysiology*, 1970, 29: 516-520.

Libourel, P.-A., Corneyllie, A., Luppi, P.-H., Chouvet, G. and Gervasoni, D. Unsupervised online classifier in sleep scoring for sleep deprivation studies. *Sleep*, 2015, 38: 815-828.

Lloyd, S. Least squares quantization in PCM. *IEEE transactions on information theory*, 1982, 28: 129-137.

Loomis, A. L., Harvey, E. N. and Hobart, G. Potential rhythms of the cerebral cortex during sleep. *Science*, 1935.

Loomis, A. L., Harvey, E. N. and Hobart, G. Cerebral states during sleep, as studied by human brain potentials. *Journal of experimental psychology*, 1937, 21: 127.

Loomis, A. L., Harvey, E. N. and Hobart, G. A. Distribution of disturbance-patterns in the human electroencephalogram with special reference to sleep. *Journal of Neurophysiology*, 1938.

Louis, R. P., Lee, J. and Stephenson, R. Design and validation of a computer-based sleep-scoring algorithm. *Journal of neuroscience methods*, 2004, 133: 71-80.

Luca, G., Haba Rubio, J., Andries, D. *et al.* Age and gender variations of sleep in subjects without sleep disorders. *Annals of Medicine*, 2015, 47: 482-491.

Maaten, L. V. D. and Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research*, 2008, 9: 2579-2605.

Mackiewicz, M., Shockley, K. R., Romer, M. A. *et al.* Macromolecule biosynthesis: a key function of sleep. *Physiological genomics*, 2007, 31: 441-457.

Magosso, E., Provini, F., Montagna, P. and Ursino, M. A wavelet based method for automatic detection of slow eye movements: A pilot study. *Medical engineering & physics*, 2006, 28: 860-875.

Malhotra, A., Younes, M., Kuna, S. T. *et al.* Performance of an automated polysomnography scoring system versus computer-assisted manual scoring. *Sleep*, 2013, 36: 573-582.

Marg, E. Development of electro-oculography: Standing potential of the eye in registration of eye movement. *AMA archives of ophthalmology*, 1951, 45: 169-185.

Marquié, J.-C., Tucker, P., Folkard, S., Gentil, C. and Ansiau, D. Chronic effects of shift work on cognition: findings from the VISAT longitudinal study. *Occup Environ Med*, 2014: oemed-2013-101993.

Martin, W., Johnson, L., Viglione, S., Naitoh, P., Joseph, R. and Moses, J. Pattern recognition of EEG-EOG as a technique for all-night sleep stage scoring. *Electroencephalography and clinical neurophysiology*, 1972, 32: 417-427.

Mayers, A. G. and Baldwin, D. S. Antidepressants and their effect on sleep. *Human Psychopharmacology: Clinical and Experimental*, 2005, 20: 533-559.

Mccarley, R. W. Neurobiology of REM and NREM sleep. *Sleep medicine*, 2007, 8: 302-330.

Mccoy, J. G. and Strecker, R. E. The cognitive cost of sleep lost. *Neurobiology of learning and memory*, 2011, 96: 564-582.

Mchugh, M. L. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 2012, 22: 276-282.

Mignot, E. Why we sleep: the temporal organization of recovery. *PLoS biology*, 2008, 6: e106.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J. and Khudanpur, S. Recurrent neural network based language model. *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

Mirowski, P. W., Lecun, Y., Madhavan, D. and Kuzniecky, R. Comparing SVM and convolutional networks for epileptic seizure prediction from intracranial EEG. *Machine Learning for Signal Processing, 2008. MLSP 2008. IEEE Workshop on*. IEEE, 2008: 244-249.

Mitchell, T. M. Machine learning. WCB. McGraw-Hill Boston, MA:, 1997.

Morgan, J. N. and Sonquist, J. A. Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association*, 1963, 58: 415-434.

Nesterov, Y. A method of solving a convex programming problem with convergence rate O (1/k2). *Soviet Mathematics Doklady*, 1983: 372-76.

Ngo, H. V., Martinetz, T., Born, J. and Molle, M. Auditory closed-loop stimulation of the sleep slow oscillation enhances memory. *Neuron*, 2013, 78: 545-553.

Ogilvie, R. D., Mcdonagh, D. M., Stone, S. N. and Wilkinson, R. T. Eye movements and the detection of sleep onset. *Psychophysiology*, 1988, 25: 81-91.

Ohayon, M. M. Epidemiology of insomnia: what we know and what we still need to learn. *Sleep medicine reviews*, 2002, 6: 97-111.

Omlin, X., Crivelli, F., Näf, M. *et al.* The Effect of a Slowly Rocking Bed on Sleep. *Scientific Reports*, 2018, 8: 2156.

Oord, A. V. D., Kalchbrenner, N. and Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Oshiro, T. M., Perez, P. S. and Baranauskas, J. A. How many trees in a random forest? *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2012: 154-168.

Pan, S.-T., Kuo, C.-E., Zeng, J.-H. and Liang, S.-F. A transition-constrained discrete hidden Markov model for automatic sleep staging. *Biomedical engineering online*, 2012, 11: 52.

Pardey, J., Roberts, S., Tarassenko, L. and Stradling, J. A new approach to the analysis of the human sleep/wakefulness continuum. *Journal of sleep research*, 1996, 5: 201-210.

Park, H.-J., Oh, J.-S., Jeong, D.-U. and Park, K.-S. Automated sleep stage scoring using hybrid rule-and case-based reasoning. *Computers and Biomedical Research*, 2000, 33: 330-349.

Pascanu, R., Mikolov, T. and Bengio, Y. Understanding the exploding gradient problem. *CoRR, abs/1211.5063*, 2012.

Pascual-Marqui, R. D., Michel, C. M. and Lehmann, D. Low resolution electromagnetic tomography: a new method for localizing electrical activity in the brain. *International Journal of psychophysiology*, 1994, 18: 49-65.

Pavlides, C. and Winson, J. Influences of hippocampal place cell firing in the awake state on the activity of these cells during subsequent sleep episodes. *Journal of Neuroscience*, 1989, 9: 2907-2918.

Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901, 2: 559-572.

Pedregosa, F., Varoquaux, G., Gramfort, A. *et al.* Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 2011, 12: 2825-2830.

Penzel, T., Zhang, X. and Fietze, I. Inter-scorer reliability between sleep centers can teach us what to improve in the scoring rules. *Journal of clinical sleep medicine: JCSM: official publication of the American Academy of Sleep Medicine*, 2013, 9: 89-91.

Piéron, H. Le problème physiologique du sommeil. *Le probléme physiologique du sommeil*, 1913.

Pop-Jordanova, N. and Pop-Jordanov, J. Spectrum-weighted EEG frequency ("brain-rate") as a quantitative indicator of mental arousal. *Prilozi*, 2005, 26: 35-42.

Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 1989, 77: 257-286.

Raizen, D. M., Zimmerman, J. E., Maycock, M. H. *et al.* Lethargus is a Caenorhabditis elegans sleep-like state. *Nature*, 2008, 451: 569-572.

Rajaratnam, S. M. and Arendt, J. Health in a 24-h society. *The Lancet*, 2001, 358: 999-1005.

Ramin, C., Devore, E. E., Wang, W., Pierre-Paul, J., Wegrzyn, L. R. and Schernhammer, E. S. Night shift work at specific age ranges and chronic disease risk factors. *Occup Environ Med*, 2015, 72: 100-107.

Rechtschaffen, A. Current perspectives on the function of sleep. *Perspectives in biology and medicine*, 1998, 41: 359-390.

Rechtschaffen, A. and Kales, A. *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*. National Institutes of Health, Bethseda, Maryland, 1968.

Resch, B. Hidden Markov Models A Tutorial for the Course Computational Intelligence. 2004.

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, 1951: 400-407.

Rochester, N., Holland, J., Haibt, L. and Duda, W. Tests on a cell assembly theory of the action of the brain, using a large digital computer. *IRE Transactions on information Theory*, 1956, 2: 80-93.

Rosenberg, R. S. and Van Hout, S. The American Academy of Sleep Medicine inter-scorer reliability program: sleep stage scoring. *Journal of clinical sleep medicine: JCSM: official publication of the American Academy of Sleep Medicine*, 2013, 9: 81.

Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 1958, 65: 386.

Roth, B. and Broughton, R. J. *Narcolepsy and hypersomnia*. Karger Basel, 1980.

Safavian, S. R. and Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 1991, 21: 660-674.

Salakhutdinov, R. and Hinton, G. Semantic hashing. *International Journal of Approximate Reasoning*, 2009, 50: 969-978.

Schaltenbrand, N., Lengelle, R. and Macher, J.-P. Neural network model: application to automatic analysis of human sleep. *Computers and Biomedical Research*, 1993, 26: 157-171.

Schlögl, A. The electroencephalogram and the Adaptive Autoregressive Model. Theory and Applications. *The Institute of Electrical and Biomedical Engineering* Graz University of Technology, Graz, Austria, 2000.

Schlögl, A. and Brunner, C. BioSig: A Free and Open Source Software Library for BCI Research. *Computer*, 2008, 41: 44-50.

Schlögl, A., Flotzinger, D. and Pfurtscheller, G. Adaptive autoregressive modeling used for single-trial EEG classification-verwendung eines Adaptiven Autoregressiven Modells für die Klassifikation von Einzeltrial-EEG-Daten. *Biomedizinische Technik/Biomedical Engineering*, 1997, 42: 162-167.

Schlögl A., A. P., M.-J. Barbanoj M.-J., Klösch G., Gruber G., Lorenzo J.L., Filz O., Koivuluoma M.,Rezek I., Roberts S.J., Värri A., Rappelsberger P., Pfurtscheller G., Dorffner G. Artefact processing of the sleep eeg in the "siesta"- Project. *1st European Medical and Biological Engineering Conference* Vienna (Austria), 1999: 1644 – 1645.

Semlitsch, H. V., Anderer, P., Schuster, P. and Presslich, O. A Solution for Reliable and Valid Reduction of Ocular Artifacts, Applied to the P300 ERP. *Psychophysiology*, 1986, 23: 695-703.

Sharpley, A., Williamson, D., Attenburrow, M., Pearson, G., Sargent, P. and Cowen, P. The effects of paroxetine and nefazodone on sleep: a placebo controlled trial. *Psychopharmacology*, 1996, 126: 50-54.

Shaw, P. J., Tononi, G., Greenspan, R. J. and Robinson, D. F. Stress response genes protect against lethal effects of sleep deprivation in Drosophila. *Nature*, 2002, 417: 287-291.

Sivaranjni, V. and Rammohan, T. Detection of sleep apnea through ECG signal features. *Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2016 2nd International Conference on*. IEEE, 2016: 322-326.

Skorucak, J. Sleep regulation: sleep restriction, extension, and microsleep detection. ETH Zurich, Ph.D. thesis DISS. ETH NO. 24844, Zurich, 2017.

Smith, J. R., Funke, W. F., Yeo, W. and Ambuehl, R. A. Detection of human sleep EEG waveforms. *Electroencephalography and clinical neurophysiology*, 1975, 38: 435-437.

Smith, J. R. and Karacan, I. EEG sleep stage scoring by an automatic hybrid system. *Electroencephalography and Clinical Neurophysiology*, 1971, 31: 231-237.

Sørensen, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, 1948, 5: 1-34.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014, 15: 1929-1958.

Stanus, E., Lacroix, B., Kerkhofs, M. and Mendlewicz, J. Automated sleep scoring: a comparative reliability study of two algorithms. *Electroencephalography and clinical neurophysiology*, 1987, 66: 448-456.

Stassen, H. Computerized recognition of persons by EEG spectral patterns. *Electroencephalography and clinical neurophysiology*, 1980, 49: 190-194.

Steinhaus, H. Sur la division des corps materiels en parties. Bull. Acad. Polon. Sci., C1. III vol IV: 801-804. 1956.

Stephenson, R., Chu, K. M. and Lee, J. Prolonged deprivation of sleep-like rest raises metabolic rate in the Pacific beetle cockroach, Diploptera punctata (Eschscholtz). *Journal of Experimental Biology*, 2007, 210: 2540-2547.

Stickgold, R. Neuroscience: a memory boost while you sleep. *Nature*, 2006, 444: 559-560.

Stickgold, R., Hobson, J. A., Fosse, R. and Fosse, M. Sleep, learning, and dreams: off-line memory reprocessing. *Science*, 2001, 294: 1052-1057.

Stone, M. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society. Series B (Methodological)*, 1974: 111-147.

Stratonovich, R. L. E. Conditional markov processes. *Theory of Probability & Its Applications*, 1960, 5: 156-178.

Sun, H., Jia, J., Goparaju, B. *et al.* Large-scale automated sleep staging. *Sleep*, 2017, 40.

Sunagawa, G. A., Séi, H., Shimba, S., Urade, Y. and Ueda, H. R. FASTER: an unsupervised fully automated sleep staging method for mice. *Genes to Cells*, 2013, 18: 502-518.

Supratak, A., Dong, H., Wu, C. and Guo, Y. DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG. *IEEE*

Transactions on Neural Systems and Rehabilitation Engineering, 2017, 25: 1998-2008.

Sutskever, I., Martens, J., Dahl, G. and Hinton, G. On the importance of initialization and momentum in deep learning. *International conference on machine learning*, 2013: 1139-1147.

Tagliazucchi, E., Von Wegner, F., Morzelewski, A., Borisov, S., Jahnke, K. and Laufs, H. Automatic sleep staging using fMRI functional connectivity data. *Neuroimage*, 2012, 63: 63-72.

Takahara, M., Kanayama, S. and Hori, T. Co-occurrence of sawtooth waves and rapid eye movements during REM sleep. *International Journal of Bioelectromagnetism*, 2009, 11: 144-148.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1996: 267-288.

Tinguely, G., Landolt, H. and Cajochen, C. Sleep habits, sleep quality and sleep medicine use of the Swiss population result. *Therapeutische Umschau. Revue therapeutique*, 2014, 71: 637-646.

Tobler, I. Evolution of the sleep process: A phylogenetic approach. *Exp. Brain Res*, 1984, 8: 207-226.

Tobler, I., Kopp, C., Deboer, T. and Rudolph, U. Diazepam-induced changes in sleep: role of the α1 GABAA receptor subtype. *Proceedings of the National Academy of Sciences*, 2001, 98: 6464-6469.

Tononi, G. and Cirelli, C. Sleep function and synaptic homeostasis. *Sleep medicine reviews*, 2006, 10: 49-62.

Trachsel, L., Dijk, D., Brunner, D., Klene, C. and Borbély, A. Effect of zopiclone and midazolam on sleep and EEG spectra in a phase-advanced sleep schedule. *Neuropsychopharmacology*, 1990, 3: 11-18.

Tsinalis, O., Matthews, P. M., Guo, Y. and Zafeiriou, S. Automatic sleep stage scoring with single-channel EEG using convolutional neural networks. *arXiv preprint arXiv:1610.01683*, 2016.

Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th international conference on Machine learning*. ACM, 2008: 1096-1103.

Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 1967, 13: 260-269.

Von Rotz, R., Kometer, M., Dornbierer, D. *et al.* Neuronal oscillations and synchronicity associated with gamma-hydroxybutyrate during resting-state in healthy male volunteers. *Psychopharmacology*, 2017, 234: 1957-1968.

Vyazovskiy, V. V. and Harris, K. D. Sleep and the single neuron: the role of global slow oscillations in individual cell rest. *Nature Reviews Neuroscience*, 2013, 14: 443-451.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. and Lang, K. J. Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 1989, 37: 328-339.

Walker, J. M. and Berger, R. J. Sleep as an adaptation for energy conservation functionally related to hibernation and shallow torpor. *Progress in brain research*, 1980, 53: 255-278.

Walter, W. G. The location of cerebral tumours by electro-encephalography. *The Lancet*, 1936, 228: 305-308.

Welch, L. R. Hidden Markov models and the Baum-Welch algorithm. *IEEE Information Theory Society Newsletter*, 2003, 53: 10-13.

Werbos, P. Beyond regression: New tools for prediction and analysis in the behavior science. *Unpublished Doctoral Dissertation, Harvard University*, 1974.

Werbos, P. J. *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*. John Wiley & Sons, 1994.

Widrow, B. and Hoff, M. E. Adaptive switching circuits. STANFORD UNIV CA STANFORD ELECTRONICS LABS, 1960.

Winkler, I., Haufe, S. and Tangermann, M. Automatic classification of artifactual ICA-components for artifact removal in EEG signals. *Behavioral and Brain Functions*, 2011, 7: 30.

Xie, L., Kang, H., Xu, Q. *et al.* Sleep drives metabolite clearance from the adult brain. *Science*, 2013, 342: 373-377.

Xu, R. and Wunsch, D. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 2005, 16: 645-678.

Yoo, S.-S., Hu, P. T., Gujar, N., Jolesz, F. A. and Walker, M. P. A deficit in the ability to form new human memories without sleep. *Nature neuroscience*, 2007, 10: 385-392.

Younes, M., Kuna, S. T., Pack, A. I. *et al.* Reliability of the American Academy of Sleep Medicine Rules for Assessing Sleep Depth in Clinical Practice. *Journal of Clinical Sleep Medicine*, 2018, 14: 205-213.

Younes, M., Raneri, J. and Hanly, P. Staging sleep in polysomnograms: analysis of inter-scorer variability. *Journal of clinical sleep medicine: JCSM: official publication of the American Academy of Sleep Medicine*, 2016, 12: 885.

Young, L.R. and Sheena, D. Eye-movement measurement techniques. *American Psychologist*, 1975, 30: 315.

Zhang, S., Zeitzer, J. M., Sakurai, T., Nishino, S. and Mignot, E. Sleep/wake fragmentation disrupts metabolism in a mouse model of narcolepsy. *The Journal of physiology*, 2007, 581: 649-663.

Zhdanova, I.V., Wang, S.Y., Leclair, O.U. and Danilova, N.P. Melatonin promotes sleep-like state in zebrafish. *Brain research*, 2001, 903: 263-268.

Curriculum Vitae

Alexander Malafeev

Born 11 June, 1989 in Vladimir, Russia
Citizen of Russia

2013-2018
PhD Student at the University of Zurich, Institute of Pharmacology and Toxicology, Zurich, Switzerland
Developing machine learning methods for automatic sleep analysis.

2012-2013
Research assistant at the University of Lugano, Institute of Computational Science, Lugano, Switzerland
Computer simulation of biomolecules.

2006-2012
Student at the Department of Physics of Lomonosov Moscow State University, Moscow, Russia
Diploma, GPA 4.75 out of 5.0
"Computer simulation of some conjugated polymer systems"

2004-2006
A.N. Kolmogorov's gymnasium, Moscow, Russia

Published papers and abstracts

Papers

The effect of a slowly rocking bed on sleep
Ximena Omlin, Francesco Crivelli, Monika Näf, Lorenz Heinicke, Jelena Skorucak, Alexander Malafeev, Antonio Fernandez Guerrero, Robert Riener, Peter Achermann
Scientific Reports volume 8, Article number: 2156 (2018)
doi:10.1038/s41598-018-19880-3

Automatic artefact detection in single-channel sleep EEG recordings
Alexander Malafeev, Ximena Omlin, Aleksandra Wierzbicka, Adam Wichniak, Wojciech Jernajczyk, Robert Riener, Peter Achermann
J Sleep Res. 2018; e12679.
doi:10.1111/jsr.12679

Structure and response to flow of the glycocalyx layer
Eduardo R. Cruz-Chu, Alexander Malafeev, Tautrimas Pajarskas, Igor V. Pivkin, and Petros Koumoutsakos.
Biophysical Journal, 106(1):232–243, 2014.

Characterization of charge-carrier transport in semicrystalline polymers: Electronic couplings, site energies, and charge-carrier dynamics in poly (bithiophene-alt-thienothiophene) [pbttt]
Carl Poelking, Eunkyung Cho, Alexander Malafeev, Viktor Ivanov, Kurt Kremer, Chad Risko, Jean-Luc Bŕedas, and Denis Andrienko.
The Journal of Physical Chemistry C, 117(4):1633–1640, 2013.

Solvated poly-(phenylene vinylene) derivatives: conformational structure and aggregation behavior
Alexander Lukyanov, Alexander Malafeev, Viktor Ivanov, Hsin-Lung Chen, Kurt Kremer, and Denis Andrienko.
J. Mater. Chem., 20:10475–10485, 2010.

Abstracts

Effects of rocking movements on sleep onset and memory performance
Ximena Omlin, Francesco Crivelli, Lorenz Heinicke, Jelena Skorucak, Alexander Malafeev, Antonio Fernandez, Robert Riener, Peter Achermann
Journal of Sleep Research 23, 255

Automatic detection of sleep episodes in long-term EEG recordings
Alexander Malafeev, Aleksandra Wierzbicka, Adam Wichniak, Peter Achermann
Journal of Sleep Research 25, 344

Automatic detection of microsleep episodes
Jelena Skorucak, David R. Schreier, Alexander Malafeev, Johannes Mathis, Peter Achermann
Journal of Sleep Research 25, 276

Automatic artefact detection in long-term single channel sleep EEG recordings

Alexander Malafeev, Ximena Omlin, Aleksandra Wierzbicka, Adam Wichniak, Wojciech Jernajczyk, Peter Achermann Journal of Sleep Research 25, 253

Awards

Best oral presentation award
Zurich Center for Integrative Physiology (ZIHP) annual meeting 2016
26.08.2016
(Automatic detection of sleep episodes in long-term EEG recordings)

Attended conferences

SSSSC 2014
Automatic artifact detection in long-term EEG recordings
Luzern (15.05.2014-16.05.2014); poster

nanotera annual meeting 2014
Automatic artifact detection in long-term EEG recordings
Lausanne (19.05.2014-20.05.2014); poster

ZIHP meeting 2014
Automatic artifact detection in long-term EEG recordings
29.08.2014
Universität Zürich; poster

Pharmacology & Toxicology POSTER DAY 2014
Automatic artifact detection in long-term EEG recordings
08.09.2014
Universität Zürich; poster

ZNZ Symposium 2014
11.09.2014
Universität Zürich; poster

Resting states and state dependent information processing in health and disease" at the Monte Verità (28.09.2014 - 01.10.2014); no poster or talk

CRPP Sleep and Health Symposium 2015
Automatic artifact detection in long-term EEG recordings

16.01.2015
Universität Zürich; poster

Nanotera annual meeting 2015
Automatic artifact detection in long-term EEG recordings
Bern (4.05.2014-5.05.2014); poster

SomnoAlert
Brussels
21-23.02.2016
Automatic detection of sleep episodes in long-term EEG recordings (talk)

Nano-Tera Annual Meeting
SwissTech Convention Center
EPFL, 25th-26th April, 2016; poster
Automatic artifact detection in long-term EEG recordings

SSSSC 2016 in Basel
28-29.04.2016
Automatic detection of sleep episodes in long-term EEG recordings; talk

ESRS 2016 in Bologna
13-16.09.*2016; two posters*

ZIHP meeting 2016
Automatic detection of sleep episodes in long-term EEG recordings
26.08.2016
Universität Zürich; talk; **best oral presentation award**

SSSSC 2017 in Lugano
12.05.2016
Automatic detection of sleep episodes in long-term EEG recordings; talk

ZNZ PhD retreat in Valens
20.05.2017
Automatic analysis of long-term EEG recordings; talk

CRPP Sleep and Health Symposium 2018 in Zurich
01.02.2018
Automatic sleep scoring; workshop talk